UNIVERSITÉ DES ANTILLES

2024 - 2026

RÉGION GUADELOUPE

MÉMOIRE PROFESSIONNEL

LA DETTE ORGANISATIONNELLE : QUAND LES RACCOURCIS D'AUJOURD'HUI DEVIENNENT LES FREINS DE DEMAIN

Milan LADAMUS
Superviseur Infrastructure et Réseaux

MIAGE

Tuteur : Thierry CELESTE - Chef de la cellule Cybersécurité
Référent : Erick STATTNER - Professeur des Universités en Informatique

2024 - 2026

Au fil de mon parcours en MIAGE à l'Université des Antilles, ma manière d'appréhender les systèmes d'information a progressivement évolué. Là où l'informatique peut d'abord être perçue à travers le développement, les infrastructures ou les outils, la formation MIAGE invite à considérer un système d'information dans son ensemble : ses composantes techniques, mais aussi les processus, les données, les acteurs, les décisions et les relations qui les relient.

Cette approche a trouvé un prolongement concret au cours de mes trois années d'apprentissage au sein de la Région Guadeloupe et plus particulièrement durant les deux années de Master où j'ai occupé le poste de superviseur infrastructure et réseaux. La transversalité de cette fonction m'a conduit à intervenir sur des problématiques variées, à la croisée du développement, de l'infrastructure, de la cybersécurité et de la gouvernance du système d'information. Elle m'a surtout appris qu'une solution technique ne peut être pleinement comprise indépendamment de l'environnement organisationnel dans lequel elle doit s'inscrire, être utilisée, maintenue et transmise.

Ce mémoire constitue l'aboutissement de ce parcours. Il prend principalement appui sur les travaux menés au sein de la Région Guadeloupe et cherche à en restituer non seulement les réalisations, mais également le cheminement : les besoins identifiés, les contraintes rencontrées, les arbitrages effectués, les réponses apportées, leurs résultats ainsi que leurs limites. Une attention particulière est accordée à la continuité des travaux engagés et aux conditions permettant aux solutions mises en place de rester compréhensibles, maintenables et réutilisables au-delà de leur contexte immédiat.

La prise de recul sur ces expériences nourrit enfin le sujet de réflexion développé dans ce mémoire : la dette organisationnelle. J'interroge plus largement la manière dont les contraintes du quotidien dictent souvent les choix effectués autour d'un système d'information et continuent à avoir des conséquences bien après leur mise en œuvre.

RÉGION GUADELOUPE

PRÉAMBULE

Milan LADAMUS
Superviseur Infrastructure et Réseaux

TABLE DES MATIÈRES

REMERCIEMENTS

Je tiens à remercier sincèrement toutes les personnes qui ont contribué, par leur accompagnement, leurs enseignements ou leurs conseils, à mon évolution au cours de ces années de formation et d'apprentissage.

Je remercie tout particulièrement Thierry CELESTE, mon maître d'apprentissage, qui m'a accompagné tout au long de ces trois années, pour la confiance qu'il m'a accordée et l'autonomie qu'il m'a progressivement laissée dans la conduite de mes missions. Nos échanges, son expérience et les responsabilités qu'il m'a confiées m'ont permis de progresser aussi bien techniquement que dans ma manière d'aborder les problématiques du système d'information.

Je remercie également Erick STATTNER, mon tuteur pédagogique, pour les orientations apportées tout au long de mon parcours en MIAGE. Plusieurs notions, méthodes et réflexes transmis au cours de la formation font aujourd'hui encore partie de ma manière de travailler professionnellement. Et plus largement, son implication et la rigueur qu'il transmet aux étudiants ont durablement contribué à ma manière d'aborder mon travail.

Mes remerciements vont aussi à Gilbert RINALDO, dont les enseignements m'ont transmis de nombreuses méthodes et façons de raisonner que je continuerai à mobiliser bien au-delà de ma formation. Ils ont fortement contribué à structurer ma manière d'analyser un problème, de choisir une démarche adaptée et de formaliser une réponse.

Je tiens également à remercier Henri ROUSSEL DUPRÉ, ses enseignements en gestion de projet, mais aussi la manière d'aborder les responsabilités, les décisions et les valeurs professionnelles ont particulièrement nourri ma réflexion et ma façon d'envisager le métier.

Plus généralement, je remercie l'ensemble des enseignants de la MIAGE qui ont contribué, au fil de mon parcours, à construire les connaissances que je mobilise aujourd'hui.

Au sein de la Région Guadeloupe, je tiens à remercier Roland AZINCOURT, pour m'avoir permis d'évoluer durant ces années au sein de la Direction de la Transition Numérique et dans un environnement offrant l'opportunité d'intervenir sur des sujets particulièrement variés.

Je souhaite également remercier Sarah VITALIS-SIMON, dont le parcours et la posture professionnelle ont constitué pour moi un exemple.

Je remercie aussi Patrice LOPES pour son engagement, sa détermination et la conviction avec laquelle il défend les sujets auxquels il croit. Son attitude et sa manière de porter ses responsabilités ont constitué pour moi une source d'inspiration.

Je remercie également Anouk ROBILLARD pour ses conseils, sa capacité à attirer mon attention sur des éléments que je n'aurais pas nécessairement identifiés seul.

Enfin, je remercie Kaëna MAUGRAN pour son aide et pour l'expertise qu'elle a régulièrement su m'apporter sur des sujets plus éloignés de mes domaines de maîtrise.

Plus largement, je remercie l'ensemble de l'équipe de la DTN et de la DOREIG de la Région Guadeloupe pour leur accueil, leur disponibilité et les nombreux échanges professionnels et humains qui ont accompagné ces années d'apprentissage.

Ces trois années ont constitué bien davantage qu'une succession de missions professionnelles. Les personnes rencontrées, les responsabilités qui m'ont été confiées, les méthodes qui m'ont été transmises et les différentes manières de penser auxquelles j'ai été confronté ont progressivement façonné la posture professionnelle que je souhaite désormais continuer à développer.

INTRODUCTION

Gagner du temps. L'expression évoque spontanément un bénéfice : avancer plus vite, répondre à une urgence ou consacrer ses ressources à ce qui paraît prioritaire. Dans une organisation, cela implique nécessairement des décisions. Tout ne peut être réalisé immédiatement, avec le même niveau d'exigence et dans les mêmes délais.

Mais que deviennent ces choix rapides avec le temps ? Un choix pertinent aujourd'hui, le reste-t-il lorsque le contexte évolue ? Et le temps gagné à court terme constitue-t-il toujours un gain à plus long terme ?

Ces interrogations ont progressivement accompagné mon parcours en apprentissage, et plus particulièrement mes deux années de Master MIAGE réalisées au sein de la Région Guadeloupe, où j'ai occupé le poste de superviseur infrastructure et réseaux. La transversalité de cette fonction m'a conduit à intervenir sur des problématiques variées de développement, d'infrastructure, de cybersécurité et de gouvernance du système d'information.

Les travaux réalisés répondaient à des besoins différents, mais partageaient un même enjeu : améliorer la maîtrise du système d'information et permettre aux solutions, aux informations et aux processus de rester exploitables dans le temps. Cette problématique professionnelle s'est notamment traduite par la reprise d'un projet d'internalisation d'un portail open data, la construction d'une cartographie puis d'un référentiel applicatif, la formalisation et l'automatisation du traitement des incidents de cybersécurité, des travaux autour de la gouvernance des identités et des accès, ainsi que l'étude d'un assistant documentaire fondé sur l'intelligence artificielle.

Ces missions m'ont confronté à des situations où la réponse technique ne suffisait pas toujours. Documentation insuffisante, informations dispersées, opérations manuelles, responsabilités à clarifier ou dépendances humaines pouvaient également conditionner la réussite et surtout la pérennité d'une solution. Certaines difficultés observées semblaient par ailleurs provenir de choix ou de compromis initialement acceptables, mais dont les conséquences s'étaient progressivement installées dans le temps.

C'est à partir de cette prise de recul qu'est né le sujet de réflexion consacré à la dette organisationnelle. Cette notion permet d'interroger les effets à long terme de certains arbitrages sans considérer pour autant que tout compromis constitue un problème.

Cette réflexion conduit à la problématique suivante :

> **Comment l'accumulation de choix à court terme transforme-t-elle le temps gagné aujourd'hui en dette organisationnelle freinant les systèmes d'information de demain ?**

Le mémoire suit ainsi une progression allant du terrain vers la réflexion : après la présentation de la collectivité et du contexte dans lequel les travaux ont été réalisés, les principales missions sont exposées à travers leurs besoins, les réponses apportées, leurs résultats et leurs limites. Deux bilans permettent ensuite d'en apprécier les apports pour la collectivité et pour mon évolution professionnelle, avant que le sujet de réflexion ne mette ces expériences en perspective à travers la dette organisationnelle.

Derrière cette progression demeure finalement une question simple :
Que devient réellement le temps qu'une organisation pense avoir gagné ?

01

PRÉSENTATION DE LA COLLECTIVITÉ

LE CONSEIL RÉGIONAL DE GUADELOUPE ET LES ENJEUX NUMÉRIQUES D'UNE GRANDE COLLECTIVITÉ

La Région Guadeloupe intervient dans des domaines très différents, allant notamment du développement économique à la formation, en passant par l'aménagement du territoire, les transports, les lycées ou encore la gestion de programmes européens. Cette diversité de compétences se traduit directement dans son organisation interne : derrière une même collectivité coexistent des métiers, des contraintes et des besoins numériques parfois très éloignés les uns des autres.

Cette pluralité se retrouve dans la structure administrative de la collectivité, organisée autour de plusieurs directions générales adjointes, directions et services spécialisés. Afin d'en donner une vision d'ensemble, l'organigramme complet de la Région Guadeloupe est présenté en annexe A (figure A.1). Il permet notamment de situer la Direction de la Transition Numérique au sein d'un environnement composé de nombreuses entités aux missions distinctes. Cette organisation implique que le système d'information doive répondre à des besoins variés tout en assurant la cohérence, la continuité et la sécurité des services numériques proposés à l'ensemble de la collectivité.

C'est dans cet environnement que s'inscrit la Direction de la Transition Numérique. Son périmètre couvre notamment l'environnement de travail numérique, les infrastructures, les projets numériques, l'accompagnement de la transition numérique territoriale ainsi que la cybersécurité. Son rôle dépasse ainsi la seule mise à disposition d'outils : elle intervient sur des ressources et des services dont dépendent quotidiennement les autres composantes de la collectivité.

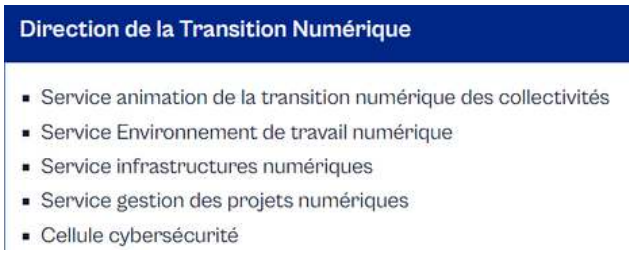
Cette transversalité constitue un élément important pour comprendre les travaux présentés dans ce mémoire. Une intervention sur une infrastructure, une application, un processus d'habilitation ou un incident de sécurité ne concerne que rarement un objet technique isolé : elle s'inscrit dans un système déjà existant, utilisé par différents acteurs et soumis à des contraintes techniques autant qu'organisationnelles.

02

2.1. POSITIONNEMENT DANS L'ORGANISATION, MISSIONS ET ORGANISATION PRÉVISIONNELLE ET EFFECTIVE DES TRAVAUX

Mon apprentissage s'est déroulé dans la **Direction de la Transition Numérique** (DTN), rattachée au **Directeur Général des Services** (DGS), au sein de la **cellule cybersécurité**. L'organigramme présenté en annexe A (figure A.1) permet de situer la Direction de la Transition Numérique ainsi que la cellule au sein de l'organisation. Celle-ci reposait alors sur un effectif réduit : mon maître d'apprentissage en constituait l'unique agent permanent, auquel je venais apporter un appui dans le cadre de mon alternance. Cette configuration m'a placé au contact direct des problématiques de cybersécurité, tout en m'amenant à intervenir plus largement sur des sujets d'infrastructure, de gouvernance et d'évolution du système d'information. La figure 2.1, présente les services au sein de cette direction.

Figure 2.1 — Services dans la Direction de La Transition Numérique

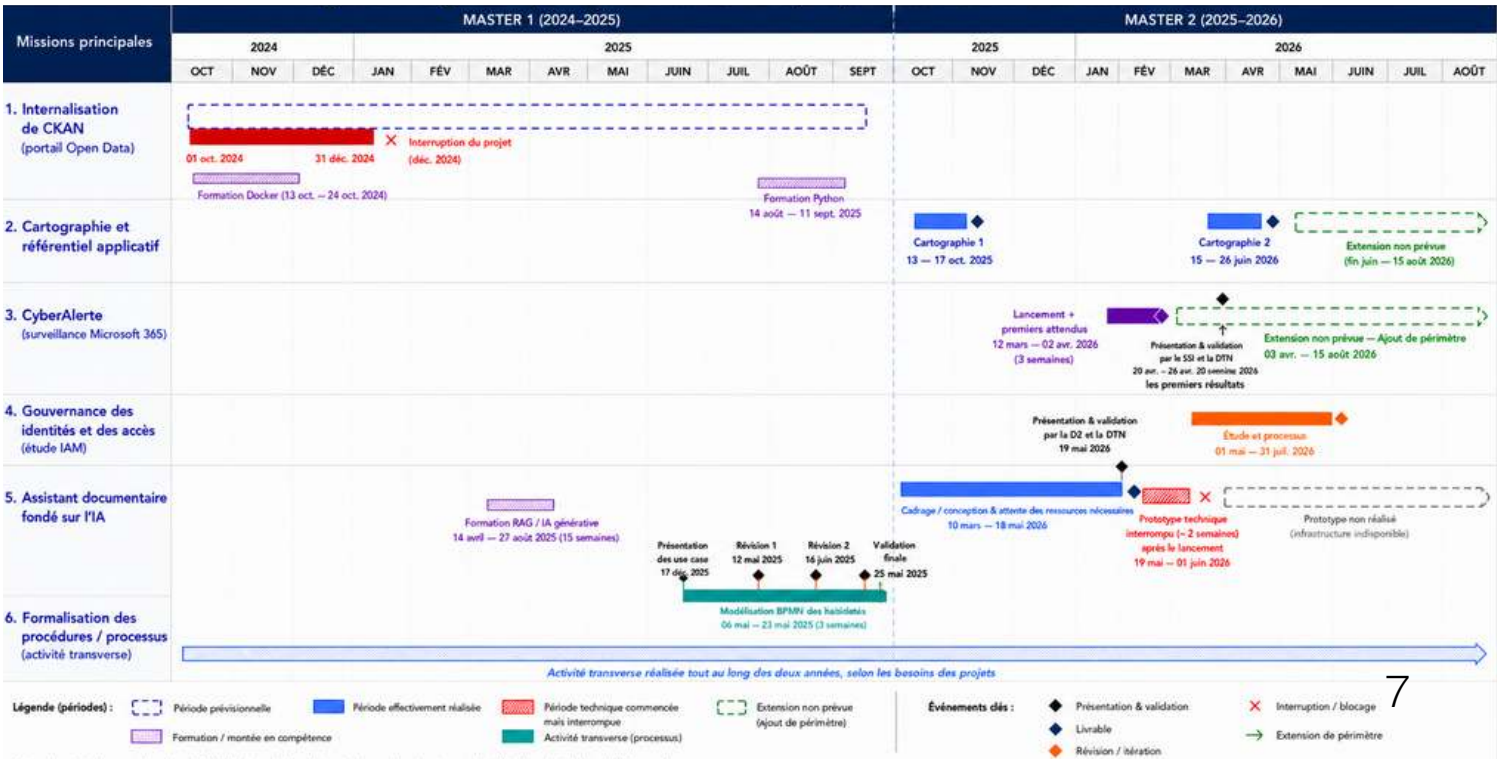


Dans ce contexte, l'autonomie s'est rapidement imposée comme mode de fonctionnement. La qualité des premiers livrables, ma rapidité d'exécution et les initiatives prises ont conduit mon maître d'apprentissage à me confier une marge de manœuvre importante. Cette confiance restait accompagnée d'échanges et de validations, tout en me permettant d'explorer un problème, de proposer une réponse et de la mener à bien.

Cette position m'a également amené à travailler avec des interlocuteurs variés : équipes techniques de la DTN, responsables de projets, prestataires ou acteurs impliqués dans la sécurité et les habilitations. Cette diversité explique en partie l'hétérogénéité des missions, dont certaines relevaient du périmètre initial de l'apprentissage tandis que d'autres ont émergé selon les besoins.

Le programme de travail a donc évolué au fil du temps : certains projets ont été approfondis, d'autres réorientés ou interrompus lorsque leur contexte ne permettait plus de les poursuivre. La figure 2.2 met en parallèle l'organisation temporelle des principales missions sur les deux années de Master et leur déroulement effectif.

Figure 2.2 — Gantt prévisionnel et effectif des principales missions au cours du Master



Afin de conserver une lecture synthétique du déroulement de l'apprentissage, le tableau 2.1 présente uniquement les principales missions qui seront développées dans la suite du mémoire. Il en restitue la situation initiale, les principales évolutions intervenues au cours de l'année, ainsi que leur état en fin de période. Une vision plus exhaustive des travaux réalisés, des décisions prises, des livrables produits et des éléments permettant d'en assurer la traçabilité est proposée dans la matrice présentée en annexe B (tableau B.1).

TABLEAU 2.1 — ÉVOLUTION DES MISSIONS CONFIÉES ENTRE PRÉVISION ET RÉALISATION

Mission	Situation initiale	Évolution au cours de l'apprentissage	Situation en fin d'apprentissage
Internalisation CKAN	Plateforme dépendante d'un hébergement externe ; projet d'internalisation engagé.	Préparation technique réalisée, puis blocage lié à une dépendance humaine.	Projet interrompu, conditions de reprise identifiées.
Référentiel applicatif	Informations applicatives dispersées et difficilement exploitables pour le pilotage.	Le besoin initial de centralisation s'est élargi vers la structuration et le suivi du patrimoine applicatif.	Référentiel conçu et démontré, avec perspectives d'enrichissement.
Traitement des incidents cyber	Traitement reposant largement sur des opérations manuelles et répétitives.	Formalisation du traitement puis automatisation de certaines actions avec PowerShell.	Automatisation fonctionnelle, à pérenniser et compléter.
Gouvernance IAM	Processus d'habilitation nécessitant davantage de formalisation, de traçabilité et de clarification des responsabilités.	Modélisation des processus, définition des besoins puis étude d'une réponse plus structurée.	Besoins et processus formalisés, démarche IAM à poursuivre.
Assistant documentaire IA	Difficulté d'accès et de transmission de certaines connaissances documentaires.	Conception et présentation d'une proposition, puis apparition de contraintes empêchant sa poursuite.	Projet arrêté au stade de l'étude, conditions de reprise identifiées.
Procédures / processus	Plusieurs processus reposaient sur des pratiques peu ou insuffisamment formalisées.	Modélisation et formalisation de procédures sur différents besoins de la collectivité.	Procédures documentées, utilisables comme support de référence et de transmission.

L'écart entre ce qui avait été envisagé et ce qui a finalement été réalisé a directement influencé plusieurs des travaux présentés dans ce mémoire. Cela permet de mieux comprendre l'environnement dans lequel ils ont été conduits et devient particulièrement instructif lorsque certains ajustements résultent de dépendances, de ressources limitées, de difficultés de coordination ou de conditions de mise en œuvre qui dépassent la seule dimension technique.

2.2. ENVIRONNEMENT TECHNIQUE ET CONTRAINTES DE RÉALISATION

Les travaux ont été menés dans un environnement technique mêlant infrastructures hébergées et solutions exploitées en interne. La collectivité s'appuie notamment sur Scaleway[1] pour certaines ressources et performances, et utilise principalement PostgreSQL[2] pour les bases de données concernées par mes travaux. Une part importante des solutions logicielles repose également sur des éditeurs ou prestataires externes. Les postes de travail sont protégés par SentinelOne[3] et Zscaler[4], ce dernier participant également à la sécurisation des accès distants, tandis que l'authentification s'appuie largement sur l'écosystème Microsoft.

À ces choix techniques s'ajoutent des contraintes propres au fonctionnement d'une collectivité. Les projets doivent composer avec les marchés publics, les budgets disponibles, les procédures de validation et la disponibilité d'interlocuteurs appartenant à différentes directions. Certaines interventions nécessitent ainsi plusieurs acteurs ou autorisations, tandis que les priorités opérationnelles peuvent conduire à réorganiser le calendrier initialement envisagé.

La disponibilité des compétences constitue également un enjeu. Au sein de la cellule cybersécurité, mon maître d'apprentissage était alors l'unique agent permanent, auquel je venais apporter un appui. Plus généralement, la concentration de certaines connaissances ou capacités d'intervention auprès d'un nombre limité de personnes peut créer des dépendances qui influencent directement la continuité et l'avancement des projets.

Enfin, les choix technologiques s'inscrivent dans une réflexion croissante autour de la maîtrise des infrastructures, des données et des dépendances envers les fournisseurs. À l'échelle de la collectivité, cette orientation se traduit notamment par la volonté de privilégier, lorsque cela est possible, des solutions françaises ou européennes et de limiter progressivement certaines dépendances envers des solutions extra-européennes. Cette orientation doit néanmoins composer avec un système d'information déjà fortement intégré à plusieurs écosystèmes américains, notamment celui de Microsoft. Il ne s'agit donc pas simplement de substituer un outil à un autre : les choix futurs doivent tenir compte des dépendances techniques, contractuelles et organisationnelles déjà constituées.

Disponibilité des acteurs, accès, responsabilités, marchés, dépendances existantes ou priorités concurrentes forment le cadre dans lequel les travaux présentés dans la suite du mémoire ont été menés.

1. Entreprise française proposant des services d'hébergement informatique, notamment des serveurs, du stockage et des infrastructures accessibles à distance
2. Système de gestion de base de données permettant de stocker, organiser et interroger des données de manière structurée.
3. Solution de cybersécurité permettant de surveiller et protéger les postes de travail et serveurs contre les menaces informatiques.
4. Solution de cybersécurité permettant de sécuriser et contrôler les accès des utilisateurs aux ressources Internet et aux applications de l'entreprise.

PRÉSENTATION DES TRAVAUX RÉALISÉS

03

3.1.1 CONTEXTE ET OBJECTIFS DU PROJET

KaruData est le portail Open Data[5] de la Région Guadeloupe. Il centralise et met à disposition des données territoriales produites par la collectivité et ses partenaires. On peut par exemple y publier un jeu de données consacré aux établissements scolaires de Guadeloupe, accompagné de sa description, de son producteur, de sa date de mise à jour et de ressources téléchargeables.

Lors de ma première année d'alternance, KaruData reposait sur une solution fournie et hébergée par OpenDataSoft. Si ce fonctionnement répondait aux besoins de publication, les ambitions autour de la plateforme évoluaient : davantage de maîtrise de l'infrastructure, plus de possibilités de personnalisation, une meilleure réversibilité[6] et une réduction de la dépendance à un prestataire externe. La question du coût récurrent renforçait également l'intérêt d'étudier une alternative.

L'hypothèse d'une internalisation a ainsi été explorée autour de CKAN (Comprehensive Knowledge Archive Network), une plateforme open source[7] spécialisée dans la publication et la gestion de catalogues de données. Elle reprend la même logique fondamentale : des organisations publient des jeux de données, eux-mêmes associés à une ou plusieurs ressources.

CKAN offrait ainsi une alternative cohérente avec les objectifs recherchés : solution open source, architecture extensible, API[8] et possibilités de personnalisation. Son internalisation impliquait toutefois un transfert de responsabilités vers la collectivité : hébergement, mises à jour, sauvegardes, supervision et maintien en conditions opérationnelles devaient désormais être assumés.

Tableau 3.1 — Principaux besoins identifiés et adéquation de CKAN

Besoin	Apport attendu de CKAN
Réduire la dépendance à un fournisseur	Solution open source pouvant être déployée et exploitée par la collectivité ou par différents prestataires
Maîtriser l'environnement technique	Choix de l'hébergement, des mises à jour, des sauvegardes et de l'organisation des environnements
Accroître la capacité d'évolution	Architecture extensible, API et système de plugins permettant des intégrations et personnalisations avancées

Une première estimation du coût total de possession (TCO[9]) sur 24 mois renforçait l'intérêt économique envers CKAN.
On retrouve un coût de 112 600 € pour le scénario externalisé contre 17 231 € estimé pour CKAN, soit une différence de coût estimée à 95 369 € (**-84,7 %**). La structure du calcul et les principales hypothèses retenues sont présentées en annexe C (figure C.1).

5. Données rendues librement accessibles afin de pouvoir être consultées, réutilisées et partagées par tous.
6. Capacité à revenir à un état antérieur ou à transférer une solution vers un autre environnement sans perte majeure de données, de fonctionnement ou de maîtrise.
7. Logiciel dont le code source est accessible et peut être consulté, modifié et redistribué selon les conditions de sa licence.
8. API (Application Programming Interface) : interface permettant à différents logiciels ou services informatiques de communiquer et d'échanger des données entre eux.
9. Coût total de possession d'une solution sur toute sa durée d'utilisation, incluant notamment l'acquisition, l'hébergement, la maintenance, l'exploitation et le temps humain mobilisé.

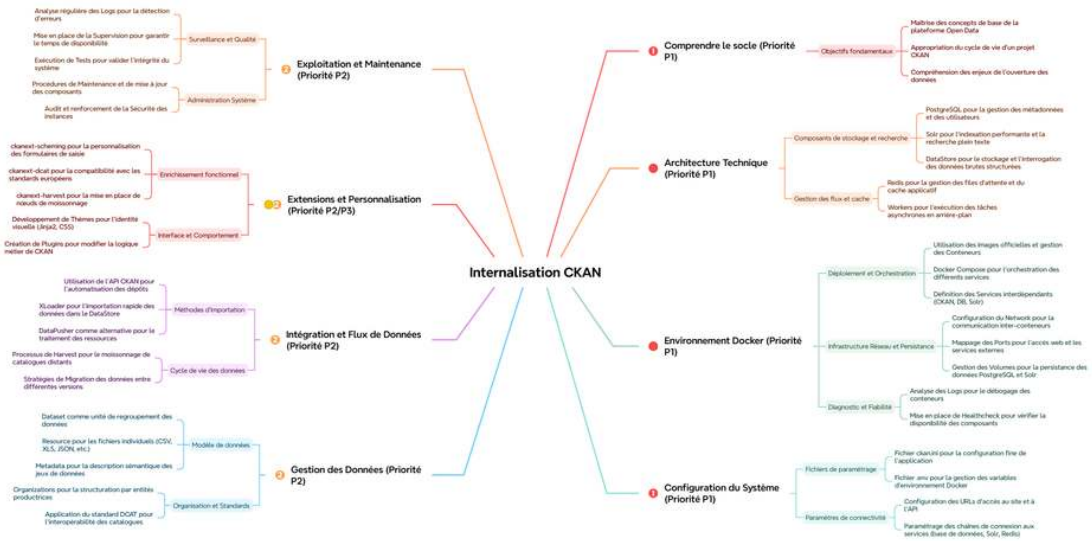
3.1.2 TRAVAUX ENGAGÉS

Lorsque j'ai repris le projet, celui-ci ne partait pas d'une page blanche. Un précédent alternant avait déjà défini une première architecture, installé CKAN sur un serveur de la collectivité et laissé une documentation permettant théoriquement de poursuivre les travaux. Avant de modifier cet existant, il fallait donc d'abord le comprendre.

Une difficulté est rapidement apparue : l'ensemble du projet reposait sur Docker[10], que je n'avais encore jamais utilisé dans un contexte opérationnel. J'ai donc commencé par une montée en compétence ciblée sur cette technologie, parallèlement à mes recherches sur CKAN autour de son architecture et des composants avec lesquels j'allais devoir interagir.

L'objectif n'était pas d'attendre de tout maîtriser avant de commencer. Il s'agissait plutôt de réduire progressivement les zones d'incertitude et d'identifier les connaissances réellement utiles à la poursuite du projet. Cette première phase se rapprochait ainsi d'un technical spike[11] : consacrer un temps limité à l'exploration d'une inconnue technique afin de pouvoir avancer par la suite avec davantage de maîtrise. Pour organiser cette montée en compétence, j'ai construit une carte mentale regroupant les notions rencontrées selon trois critères : leur priorité, mon niveau de maîtrise et leur utilité potentielle pour la suite du projet. On y retrouvait notamment Docker pour la conteneurisation, les réseaux et les volumes, PostgreSQL pour la gestion des données, Solr pour l'indexation et la recherche, Redis pour le stockage temporaire en mémoire, le DataStore pour l'exploitation des données tabulaires, ainsi que les API CKAN. Ces notions occupaient des positions différentes dans la carte selon leur importance immédiate pour la poursuite du projet.

Figure 3.1 — Carte mentale des connaissances identifiées pour la reprise du projet CKAN



Une fois les fondamentaux acquis, j'ai rapidement basculé vers une logique d'apprentissage par la pratique. Le projet devenait lui-même un support d'apprentissage : les notions étudiées pouvaient être testées immédiatement sur l'environnement de développement, tandis que les difficultés rencontrées indiquaient ce qu'il me restait à approfondir.

Néanmoins, c'est à ce moment qu'est apparu le premier véritable problème.
Malgré le respect de la procédure laissée par mon prédécesseur, CKAN restait inaccessible depuis le navigateur. Aucun message d'erreur directement exploitable ne permettait de déterminer si la cause se situait dans CKAN, Docker, l'un de ses services dépendants ou dans son exposition réseau.

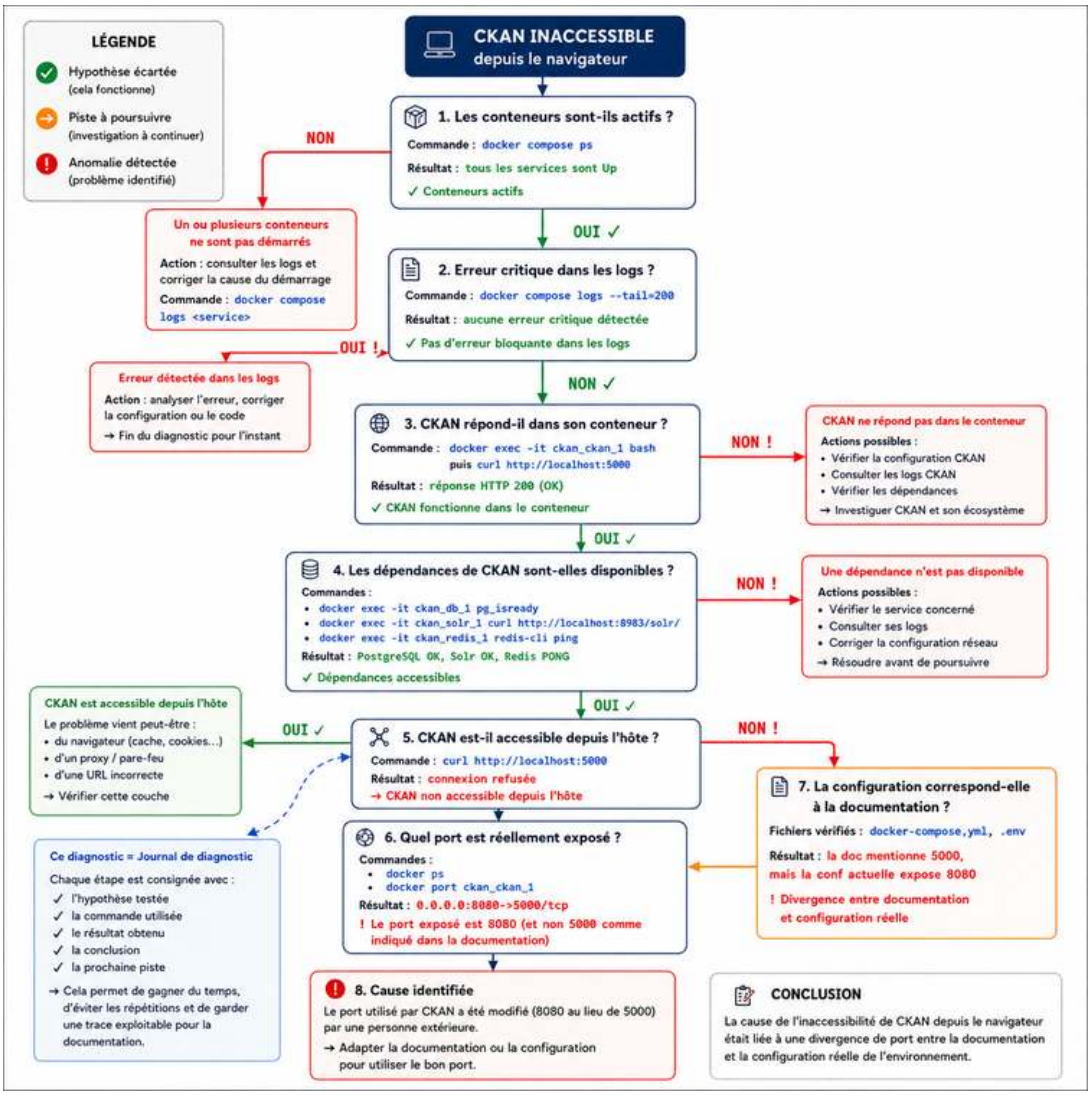
Plutôt que de multiplier les essais, j'ai organisé l'investigation selon une logique d'isolation progressive de la panne. À partir du symptôme initial, chaque vérification devait permettre soit d'écarter une famille de causes, soit de déterminer la prochaine couche à examiner.

Pour matérialiser ce raisonnement, j'ai construit un arbre de diagnostic évolutif. Chaque nœud associait une hypothèse, une vérification, les commandes exécutées et le résultat observé. L'arbre faisait simultanément office de journal de diagnostic : il conservait l'historique des pistes déjà explorées, évitait de répéter les mêmes essais et facilitait mes recherches documentaires en me permettant d'éviter d'investiguer sur des solutions déjà testées.

10. Plateforme permettant d'exécuter des applications dans des environnements isolés et reproductibles, appelés conteneurs.

11. Courte phase d'exploration technique visant à tester la faisabilité d'une solution, réduire une incertitude ou acquérir rapidement les connaissances nécessaires avant son développement.

Figure 3.2 — Arbre de diagnostic évolutif utilisé pour rechercher la cause de l'indisponibilité de CKAN



Cette historisation présentait également un intérêt au-delà de l'incident lui-même : une fois enrichi, l'arbre pouvait servir de base à une documentation de dépannage pour un futur intervenant. Le diagnostic produisait ainsi non seulement une réponse ponctuelle, mais également une connaissance réutilisable.

Progressivement, plusieurs hypothèses ont été écartées. Les conteneurs étaient actifs, aucune erreur bloquante n'apparaissait dans leurs journaux et les principaux composants répondaient correctement. L'enquête s'est alors resserrée sur l'exposition du service vers l'hôte. La vérification des ports réellement publiés par Docker a finalement révélé une divergence entre la configuration effective de l'environnement et celle décrite dans la documentation.

Cette différence, en apparence mineure, allait révéler un problème dépassant largement le seul diagnostic technique.

3.1.3. DÉPENDANCE HUMAINE ET INTERRUPTION DU PROJET

L'investigation a finalement montré que le port utilisé par l'environnement CKAN avait été modifié manuellement par un agent sans que cette modification soit répercutée dans la documentation disponible.

Deux phénomènes permettent de caractériser cette situation. Le premier, « configuration drift » désigne l'écart qui apparaît entre l'état réel d'un environnement et son état de référence, tandis que le « documentation drift » apparaît lorsque la documentation ne reflète plus fidèlement cet environnement.

Le changement de port n'était donc pas problématique en lui-même. C'est surtout son absence de traçabilité qui avait transformé une modification technique relativement simple en une recherche beaucoup plus longue. Une fois cette difficulté corrigée, la reprise du projet semblait possible. Pourtant, un second obstacle est apparu presque immédiatement : mes accès au serveur ont été supprimés par un interlocuteur.

En effet, ce serveur constituait l'environnement prévu pour le projet avec toutes les ressources nécessaires à l'exécution de CKAN et de ses différents composants. Utiliser un autre environnement ne constituait pas non plus une solution immédiate, puisque toute nouvelle installation nécessitait une validation par le même interlocuteur disposant des habilitations nécessaires.

Cependant, malgré plusieurs relances, la situation est restée bloquée pendant plusieurs mois.

Le problème prenait alors une autre dimension : la poursuite de l'internalisation dépendait désormais entièrement de l'intervention d'un acteur unique. Son indisponibilité suffisait à empêcher aussi bien la restauration de l'environnement existant que la mise en place d'une solution alternative.

Faute de pouvoir rétablir les conditions nécessaires à la poursuite des travaux, le projet d'internalisation a finalement été annulé. KaruData a donc continué au travers de la solution OpenDataSoft.

3.1.4. LIMITES ET ENSEIGNEMENTS

L'étude initiale mettait naturellement en avant les bénéfices attendus de CKAN : réduction des coûts, meilleure maîtrise de l'environnement, réversibilité ou encore capacité de personnalisation. Le déroulement du projet a toutefois conduit à relire ces mêmes avantages sous un autre angle.

Internaliser ne supprime pas les dépendances : cela en transforme certaines et transfère à l'organisation de nouvelles responsabilités. Le tableau 3.1 vu précédemment peut ainsi être relu différemment, comme le démontre le tableau 3.2.

Tableau 3.2 — Relecture des bénéfices attendus de CKAN avec responsabilités transférées

Bénéfice attendu	**Point de vigilance révélé par le projet**
Réduction de la dépendance à l'éditeur	L'autonomie technique suppose de disposer durablement des compétences nécessaires à l'exploitation
Meilleure réversibilité	Documentation, sauvegardes et maîtrise des configurations conditionnent la capacité réelle à reprendre l'environnement
Réduction du TCO	L'infrastructure, le MCO et le temps humain doivent rester intégrés au coût réel de l'internalisation
Maîtrise technique	La liberté de modifier l'environnement devient une fragilité si les changements ne sont ni tracés ni documentés
Personnalisation et intégration au SI	Les configurations spécifiques et les habilitations nécessaires doivent pouvoir être maintenues et partagées dans le temps

Deux fragilités ressortent particulièrement de cette expérience.

La première concerne la capitalisation de la connaissance : une modification technique non documentée avait créé un écart entre l'état réel de l'environnement et l'état connu par son nouvel intervenant.

La seconde concerne la concentration d'une capacité critique. La poursuite du projet reposait sur un interlocuteur unique, créant un « single point of failure[12] organisationnel » : l'indisponibilité d'un seul acteur suffisait à bloquer le processus.

Bien que l'internalisation n'ait finalement pas été menée jusqu'à son terme, la mission n'a pas été sans résultat. Elle a permis de reprendre et comprendre l'environnement existant, d'acquérir les compétences nécessaires à son exploitation, de résoudre le dysfonctionnement initial, de formaliser une démarche de diagnostic réutilisable et surtout d'identifier plusieurs prérequis organisationnels indispensables à une éventuelle reprise du projet.

L'expérience CKAN montre ainsi que le démarrage d'un projet dépend également de la capacité de l'organisation à conserver, transmettre et partager les connaissances, les configurations et les responsabilités nécessaires à son fonctionnement.

Cette question de la maîtrise de l'information ne concernait toutefois pas uniquement KaruData. À une échelle plus large, elle se retrouvait également dans la connaissance même du patrimoine applicatif de la collectivité, conduisant à un second chantier : la construction d'une cartographie et d'un référentiel applicatif du système d'information.

12. Situation dans laquelle une activité ou un processus dépend excessivement d'une seule personne, compétence ou responsabilité, créant un risque de blocage en cas d'indisponibilité

PRÉSENTATION DES TRAVAUX RÉALISÉS

03

3.2.1. CONTEXTE, BESOINS ET OBJECTIFS

Une application métier n'existe jamais complètement seule. Elle repose sur des serveurs, des flux, des ports, des environnements et parfois sur d'autres applications dont elle dépend. Savoir qu'une application existe ne suffit donc pas : encore faut-il connaître où elle se trouve, à quoi elle sert, de quoi elle dépend et ce qui dépend d'elle.

Cette connaissance constitue notamment un prérequis pour la cybersécurité et la gestion des risques. Le NIST Cybersecurity Framework 2.0 recommande de maintenir un inventaire des logiciels, services et systèmes de l'organisation et de les gérer en fonction de leur importance pour les objectifs métier et la stratégie de risque. Le CIS place également l'inventaire et le contrôle des actifs logiciels parmi ses contrôles fondamentaux.

Dans la pratique, cette connaissance permet de répondre à des questions très concrètes : quelles applications sont critiques ? Sur quels équipements reposent-elles ? Quels services seraient affectés par l'indisponibilité d'un composant ? Quelles applications doivent être traitées en priorité lors d'un incident ou d'une opération de maintenance ?

Or, au sein de la collectivité, cette information existait mais restait dispersée entre plusieurs outils, documents et personnes. Il n'existait pas de vue immédiatement exploitable regroupant les principales applications et leurs caractéristiques essentielles.

Une première mission de cartographie du patrimoine applicatif m'a donc été confiée, avec une contrainte forte : produire une première version en une semaine. L'objectif était de retrouver, consolider et structurer suffisamment d'informations.

Cette mission s'est présentée comme première démarche pour rendre le système d'information plus lisible. Trois besoins se sont progressivement dégagés : centraliser l'information, la fiabiliser et préparer sa mise à jour dans le temps.

3.2.2. COLLECTE, CONSOLIDATION ET CARTOGRAPHIE DE L'EXISTANT

La principale difficulté ne résidait pas dans la représentation des applications, mais dans la reconstruction de l'information nécessaire pour les représenter.

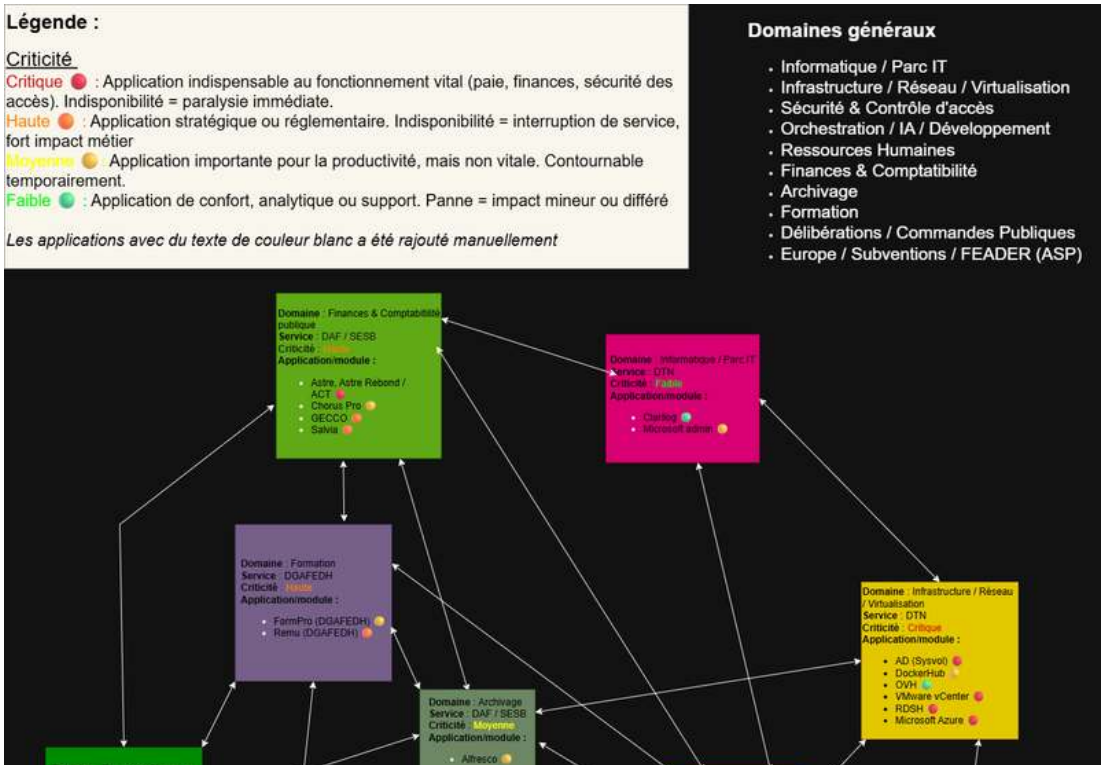
Zscaler constituait alors l'une des principales sources disponibles. L'outil fournissait plusieurs informations techniques sur le parc : serveurs, adresses IP, segments, ports, environnements ou encore niveaux de criticité. Cette représentation restait cependant principalement orientée infrastructure, alors que le besoin portait sur les applications utilisées par les métiers. Il fallait donc rapprocher progressivement deux lectures du même système d'information.

Compte tenu du délai, j'ai commencé par consolider les données disponibles dans un fichier Excel. Plusieurs agents ont parallèlement été interrogés afin d'identifier les applications effectivement utilisées. Leurs noms métier ont ensuite été rapprochés des informations techniques à partir des URL, des serveurs connus et des connaissances détenues par différents interlocuteurs.

À partir du fichier principal, une vue plus synthétique a ensuite été construite autour des informations présentant la plus forte valeur pour la cartographie. Ces éléments, étant confidentiels, ne seront pas présentés en annexe.

J'ai également expérimenté une représentation en ArchiMate, langage de modélisation d'architecture d'entreprise, permettant de représenter les relations entre différentes composantes d'un système d'information.

Figure 3.3 — Vue synthétique ArchiMate exploratoire du patrimoine applicatif



Cette représentation (voir annexe D, figure D.1 pour sa version complète) a toutefois rapidement montré une limite : toutes les dépendances n'étaient pas observables dans les sources disponibles. La majorité devait être reconstituée à partir de documentations ou d'informations partielles. La vue produite devait donc être considérée comme exploratoire et non exhaustive.

Cette première mission faisait surtout apparaître une faiblesse plus structurelle : la connaissance pouvait être reconstruite, mais au prix de nombreuses recherches manuelles, de rapprochements entre fichiers et de sollicitations auprès de différents interlocuteurs.

Un inventaire ponctuel ne suffisait donc pas. Il fallait pouvoir conserver et faire évoluer cette connaissance dans le temps.

3.2.3. CONCEPTION DU RÉFÉRENTIEL APPLICATIF

Plusieurs mois après cette première cartographie, une nouvelle demande de mon maître d'apprentissage a fait réapparaître le même besoin sous une autre forme : produire un tableau présentant la disparité des versions logicielles présentes sur le parc informatique.

Cette fois, SentinelOne constituait une source plus directement exploitable. Son inventaire associait les logiciels détectés aux équipements concernés et conservait les différentes versions observées. Le problème ne résidait donc plus dans l'absence de données, mais dans leur granularité : l'extraction représentait plus de 4 000 entrées logicielles, difficiles à interpréter à l'échelle du système d'information.

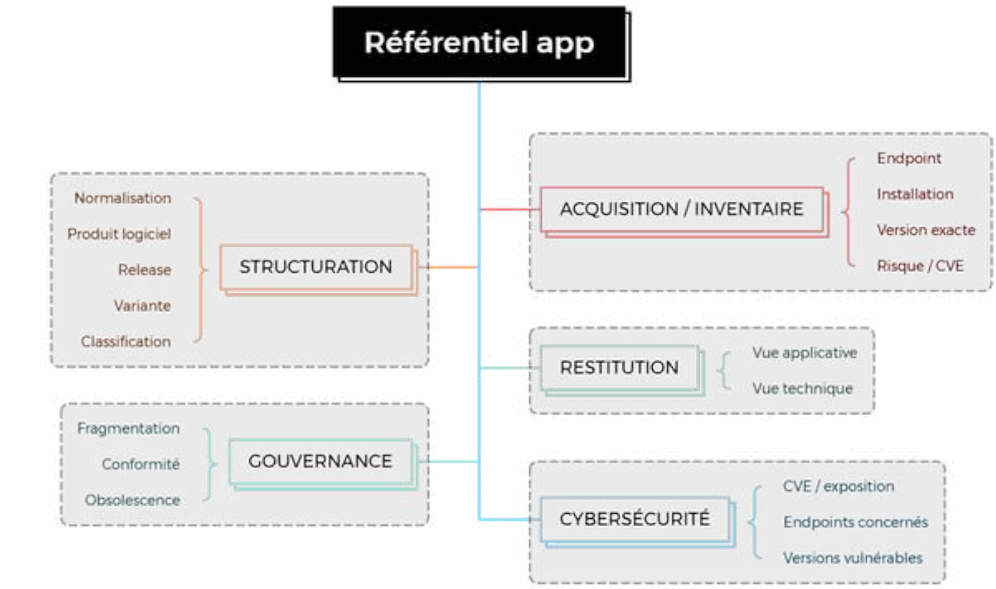
Par exemple, Adobe Photoshop 2019 et Adobe Photoshop 2022 étaient distingués, chacune de ces éditions pouvant comporter plusieurs versions. De même, Google Chrome et Google Chrome x86 apparaissaient séparément. Des runtimes, agents ou composants Linux étaient par ailleurs inventoriés au même niveau qu'un logiciel utilisateur, alors qu'ils ne répondaient pas aux mêmes usages.

SentinelOne fournissait donc une vision technique détaillée, mais celle-ci nécessitait un retraitement humain pour devenir une information consolidée. Réitérer manuellement l'export demandé était possible, mais revenait à reproduire la difficulté déjà rencontrée lors de la première cartographie. J'ai donc proposé de dépasser le besoin ponctuel pour construire une réponse réutilisable : structurer et normaliser automatiquement le patrimoine applicatif afin d'en faciliter l'exploitation dans le temps.

Structurer plusieurs niveaux de lecture

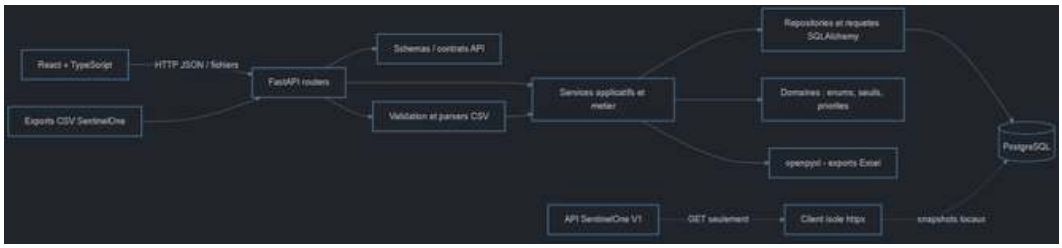
Le référentiel a été conçu autour de quatre fonctions :
collecter → transformer → conserver → restituer

Figure 3.4 — Architecture fonctionnelle du référentiel applicatif



Le backend reposait sur Python et FastAPI, PostgreSQL assurait la persistance et une interface web permettait la consultation et l'administration des données. Les observations provenant de SentinelOne étaient d'abord conservées dans une zone intermédiaire de staging[13], afin de distinguer la donnée effectivement collectée de son interprétation ultérieure.

Figure 3.5 — Architecture choisie du référentiel applicatif



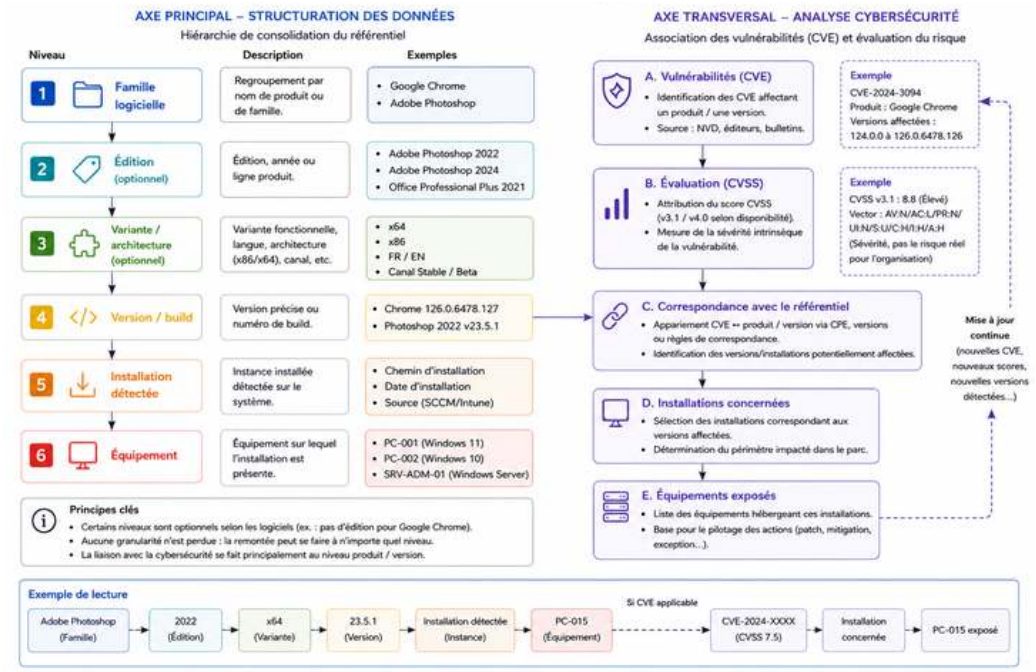
Répartition des responsabilités

- Les routers FastAPI dans `backend/app/api/` portent le protocole HTTP : paramètres, codes d'erreur, injection de session et réponses.
- Les services dans `backend/app/services/` portent les traitements : import, normalisation, consolidation, calcul de priorité, agrégation, lecture et export.
- Les repositories encapsulent les accès de persistance réutilisables.
- Les modèles SQLAlchemy dans `backend/app/models/` représentent le schéma. persistant et ses contraintes.
- Le domaine dans `backend/app/domain/` centralise les enums, constantes, seuils de scoring et structures de calcul.
- Le front affiche les données et appelle les API.

13. Zone intermédiaire dans laquelle les données brutes collectées sont temporairement stockées avant validation, transformation ou intégration dans le modèle cible.

Adobe Photoshop 2019 et Adobe Photoshop 2022 pouvaient ainsi rester deux éditions distinctes tout en appartenant à la famille Adobe Photoshop. De même, l'architecture x86 de Chrome pouvait être conservée comme caractéristique d'une variante sans empêcher une vue consolidée de la famille Google Chrome.

Figure 3.6 — Vue fonctionnelle du référentiel applicatif



Une classification complémentaire distinguait également plusieurs natures d'actifs : application métier, logiciel utilisateur, runtime, agent, composant système ou package technique. Le référentiel pouvait ainsi fournir plusieurs niveaux de lecture du même patrimoine sans supprimer l'information technique d'origine. Les fonctions principales de la V0 sont présentées en annexe D (tableau D.1)

Normaliser progressivement

La normalisation constituait le principal défi technique. Il ne s'agissait pas simplement de supprimer des doublons mais d'effectuer une forme de résolution d'entités : déterminer à quelle famille, édition ou variante rattacher chaque observation tout en conservant ses caractéristiques utiles.
L'approche retenue suivait le pipeline suivant :
prétraitement → règle connue → rapprochement → validation si nécessaire → mémorisation

Des fonctions Python homogénéisaient d'abord les valeurs : casse, espaces, et certains motifs connus : `_clean_value()` supprime les espaces en début/fin et réduit les suites d'espaces à un seul espace. `_match_key()` décode les entités HTML avec `html.unescape()`, puis applique `casefold()` pour comparer sans tenir compte de la casse. NormalizationService.from_db(session) récupère dans la base de données les règles de normalisation validées, triées par identifiant. NormalizationService.normalize() applique ensuite ces règles pour normaliser les données.

Lorsqu'une règle existait, elle était appliquée automatiquement, en privilégiant d'abord les correspondances exactes puis les préfixes les plus spécifiques, avec contrôle de l'éditeur lorsque nécessaire. À défaut, le moteur appliquait des heuristiques déterministes sur le nom et l'éditeur. Les cas incertains étaient classés AMBIGUOUS ou UNMATCHED puis soumis à correction humaine.
La correction validée était enregistrée comme règle manuelle, tracée dans un journal d'audit et réutilisée lors des analyses suivantes. La logique Human-in-the-loop[14] retenue était donc :
automatiser les cas certains → corriger les cas ambigus → enregistrer la décision → la réutiliser

Lors des premiers essais, environ 18,4 % des observations pouvaient être reconnues automatiquement à partir des seules règles initiales. L'objectif était de réduire progressivement le nombre de décisions humaines nécessaires sans multiplier les rapprochements erronés.
Des tests de non-régression ont été réalisés : les classifications déjà validées étaient également intégrées aux tests afin de détecter d'éventuelles régressions lors de l'évolution des règles.

14. Approche dans laquelle un système automatisé sollicite une intervention humaine pour valider, corriger ou arbitrer les cas incertains.

D'un export ponctuel à une connaissance actualisable

L'utilisation de l'API SentinelOne permettait également d'envisager une synchronisation régulière plutôt qu'une succession d'exports manuels. Le module Python récupérait les observations, gérait leur pagination, alimentait la zone de staging, puis déclenchait les traitements de classification et de normalisation.

Cette automatisation changeait la nature du livrable : Excel produisait une photographie ponctuelle, tandis que le référentiel pouvait suivre l'évolution du parc. Une nouvelle version pouvait être détectée, une variante soumise à qualification et la date de dernière observation utilisée pour identifier les informations susceptibles de devenir obsolètes.

L'interface était ainsi organisée autour des usages : tableau de bord, familles logicielles, éditions et versions, équipements associés et file de normalisation.

Cette structuration permettait également d'envisager de nouveaux indicateurs : nombre de versions par famille, niveau de fragmentation du parc ou, à terme, conformité à une version de référence. L'inventaire ne servait alors plus uniquement à savoir ce qui était installé, mais commençait à révéler le niveau de maîtrise de l'évolution du parc logiciel dans le temps.

Faire converger inventaire et cybersécurité

Cette structuration pouvait également être rapprochée d'un autre travail déjà mené dans l'équipe : un stagiaire avait développé un mécanisme permettant de signaler l'apparition de nouvelles CVE[15] (Common Vulnerabilities and Exposures) critiques.

Plutôt que de reconstruire cette fonctionnalité, le référentiel pouvait lui apporter le contexte manquant :
CVE → produit et versions affectés → famille normalisée → versions réellement observées → équipements concernés → criticité métier
L'association future des produits à des identifiants CVE pouvait également faciliter ce rapprochement.

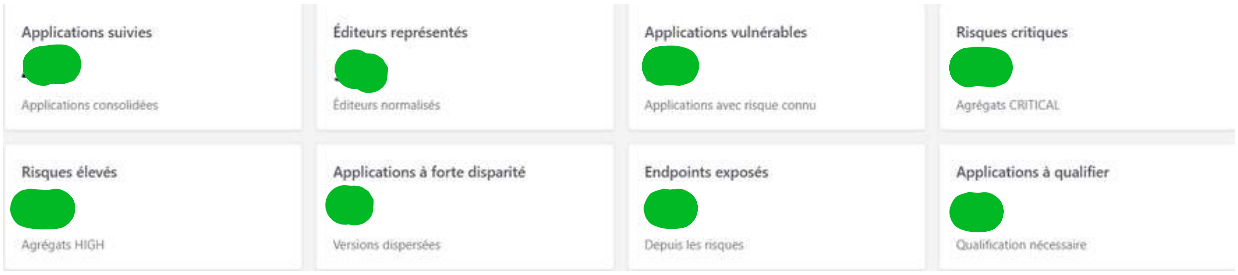
Une alerte ne signifierait alors plus uniquement : « une vulnérabilité critique concerne ce produit »
mais potentiellement : « cette vulnérabilité concerne telle version effectivement observée sur tels équipements du système d'information ».

La mission dépassait ainsi progressivement la demande initiale de comparaison de versions. SentinelOne restait la principale source d'observation du parc ; le référentiel apportait la taxonomie, la consolidation et le contexte nécessaires pour transformer cet inventaire technique en information exploitable.

3.2.4. DÉMONSTRATION ET ÉVALUATION DU RÉFÉRENTIEL

La conception du référentiel n'avait d'intérêt que si les données produites devenaient effectivement plus lisibles, plus fiables et réutilisables que les exports à l'origine de la mission.
Sur l'application, plusieurs indicateurs permettaient de prendre connaissance de l'état du parc dès le tableau de bord : familles logicielles recensées, nombre de versions observées, éléments restant à normaliser ou encore dernières synchronisations.

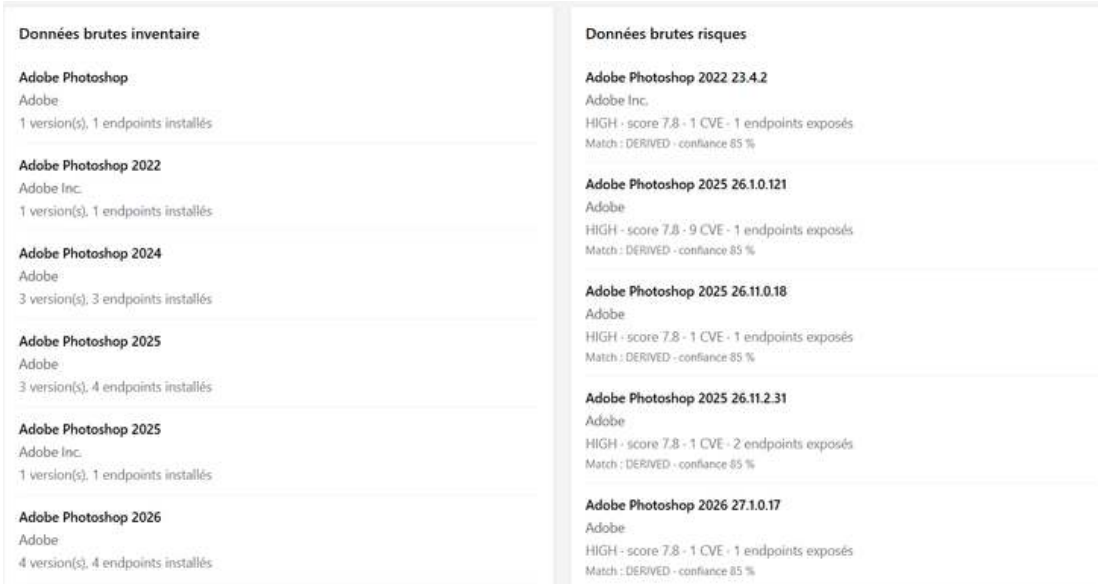
Figure 3.7 — Tableau de bord du référentiel applicatif conçu, valeurs anonymisées



15. Identifiant standard attribué à une vulnérabilité de sécurité connue afin de pouvoir la référencer de manière unique.

À partir de cette vue, l'utilisateur pouvait descendre progressivement dans le niveau de détail. Une recherche sur Google Chrome, par exemple, ne retournait plus plusieurs entrées indépendantes mais une famille consolidée conservant ses variantes et versions.

Figure 3.8 — Fiche synthétique d'une famille logicielle et répartition des variantes

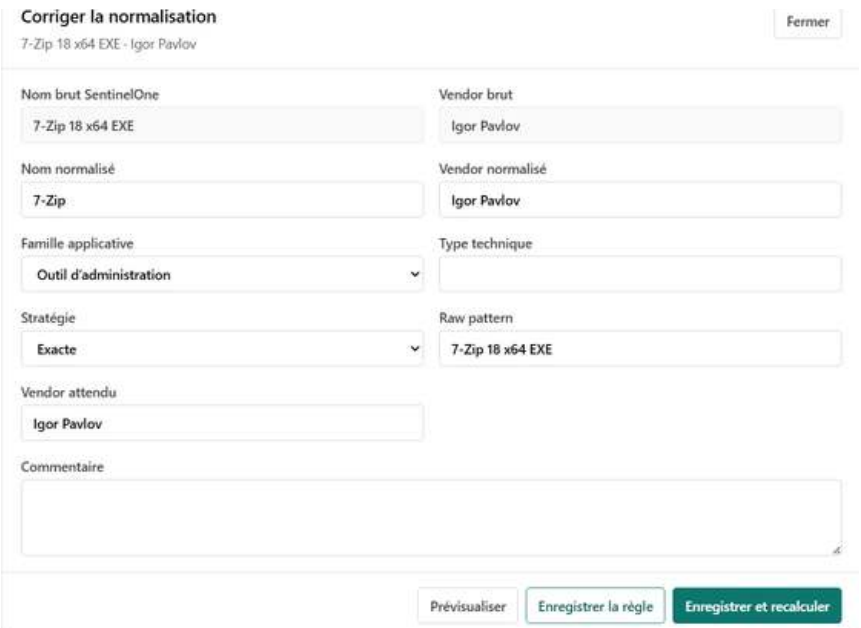


L'utilisateur pouvait ensuite sélectionner une version donnée et retrouver les équipements sur lesquels elle avait été observée ainsi que sa date de dernière détection.
La navigation suivait ainsi la logique : parc → famille → variante ou édition → version → équipements.
Une interrogation nécessitant auparavant l'exploitation manuelle d'un export pouvait donc être formulée directement dans le référentiel :
Quels équipements utilisent encore telle version d'Adobe Photoshop ?

Cela fait écho aux enseignements MIAGE d'urbanisation du système d'information, qui soulignent l'importance de disposer d'une vision structurée du patrimoine applicatif afin de pouvoir identifier rapidement ce qui existe, où cela se trouve et quels éléments sont concernés par une évolution ou un risque.

La normalisation disposait parallèlement d'une interface dédiée. Lorsqu'une observation ne pouvait pas être classée avec suffisamment de certitude, elle restait dans une file de validation plutôt que d'être automatiquement fusionnée avec une autre application.

Figure 3.9 — Interface de validation des propositions de normalisation



Une fois la décision confirmée ou corrigée, elle pouvait être réutilisée lors des synchronisations suivantes. L'interface servait à corriger à la fois l'information mais aussi à capitaliser les arbitrages réalisés.

Évaluer la fiabilité de la normalisation

Cette simplification soulevait néanmoins une question essentielle : la représentation produite restait-elle suffisamment fidèle pour servir de base à l'analyse du parc ?

Un jeu de référence de 500 observations issues de SentinelOne a été classifié manuellement en renseignant la famille attendue, l'éventuelle édition ou variante et la version. Deux traitements ont ensuite été comparés : une baseline reposant essentiellement sur le nettoyage et la proximité des chaînes de caractères, puis le pipeline complet du référentiel.

Tableau 3.3 — Comparaison de la baseline et du pipeline de normalisation

Indicateur	Baseline	Pipeline retenu
Famille correctement identifiée	82,4 %	97,6 %
Variante ou édition correctement identifiée	61,8 %	95,2 %
Version correctement conservée	93,6 %	98,8 %
Classification entièrement correcte	58,6 %	94,2 %
Faux regroupements	6,8 %	0,8 %
Fausses séparations	9,4 %	2,2 %

Une attention particulière a été accordée aux faux regroupements, puisqu'une association erronée peut être plus problématique qu'un élément temporairement laissé à qualifier, notamment dans la perspective d'un rapprochement ultérieur avec des vulnérabilités.

Le but était d'avoir une automatisation progressive et suffisamment fiable.

Mesurer la capitalisation des décisions

Le second enjeu concernait la capacité du référentiel à éviter que les mêmes arbitrages soient répétés à chaque synchronisation.

Le taux de traitement automatique est passé de 18,4 % lors de l'initialisation à 59,6 % après deux cycles d'enrichissement des règles.

Tableau 3.4 — Évolution du taux de traitement automatique après enrichissement des règles

Cycle	Traitement automatique	Reste à traiter ou qualifier
Initialisation	18,4 %	81,6 %
Après un premier cycle	34,8 %	65,2 %
Après un second cycle	59,6 %	40,4 %

La seconde colonne correspond au complément du taux de traitement automatique ; elle regroupe les observations nécessitant encore une qualification ou une intervention.

Le fonctionnement suivait ainsi continuellement la logique :
identifier → décider → mémoriser → réutiliser

Cette capitalisation était complétée par des tests de non-régression. Sur un jeu de 184 situations précédemment validées, l'ajout de nouvelles règles avait initialement provoqué plusieurs régressions. Celles-ci ont été détectées puis corrigées avant intégration, permettant au jeu final de retrouver les 184 résultats attendus sur 184.

Éprouver le référentiel face à l'augmentation du volume

Le pipeline complet a enfin été exécuté sur plusieurs jeux de données à l'aide de la fonction time.perf_counter() de la bibliothèque standard Python.

Tableau 3.5 — Benchmark de traitement du pipeline

Équipements	Installations traitées	Durée totale	Débit calculé	Erreurs
100	6 420	3,1 s	2 071 inst./s	0 %
500	31 870	5,0 s	6 374 inst./s	0 %
1 000	63 920	9,8 s	6 522 inst./s	0 %
2 500	159 480	11,5 s	13 868 inst./s	0 %

Sur le périmètre testé, aucune saturation ni erreur de traitement n'a été observée jusqu'à 159 480 installations. Le débit calculé augmente avec le volume, ce qui suggère notamment que certains coûts fixes du traitement sont progressivement amortis. Ces résultats doivent cependant être interprétés avec prudence : la non-linéarité observée justifie de privilégier plusieurs répétitions et un environnement de test parfaitement contrôlé avant de conclure plus précisément sur le comportement à grande échelle. Les protocoles, scripts et jeux de données utilisés sont conservés en annexe D.

Les évaluations apportaient ainsi trois niveaux de validation complémentaires : fiabilité de la normalisation, capitalisation progressive des arbitrages et capacité à traiter une volumétrie croissante.

À mesure que le référentiel gagnait en lisibilité, il m'a conduit à dépasser la seule logique de consultation du parc pour m'interroger sur ce que ce type d'outil permet réellement de révéler à une organisation. Comme une carte dont on augmente progressivement la résolution, une représentation plus précise du système d'information ne fait pas seulement apparaître davantage de détails : elle met aussi en évidence des irrégularités, jusque-là difficiles à percevoir.

3.2.5. RÉSULTATS, RISQUES ET ENSEIGNEMENTS

La coexistence de nombreuses versions d'une même application constitue, par exemple, une situation fréquemment rencontrée dans les parcs informatiques. Cette diversité n'est pas nécessairement problématique : plusieurs éditions ou configurations peuvent être légitimes. Le risque apparaît davantage lorsqu'elle résulte d'une accumulation de mises à jour partielles, de logiciels devenus obsolètes, d'exceptions ou d'installations successives insuffisamment maîtrisées. Il devient alors plus difficile de déterminer les versions encore supportées, d'identifier précisément les équipements concernés par une vulnérabilité et de vérifier qu'une opération corrective a effectivement couvert l'ensemble du périmètre attendu.

L'incident d'Equifax en 2017 illustre l'importance de cette connaissance. Les attaquants ont exploité une vulnérabilité connue d'Apache Struts. L'enquête du Government Accountability Office a notamment montré que le composant vulnérable n'avait pas été correctement identifié sur le portail concerné lors de la campagne de correction. Ce cas rappelle qu'identifier une CVE ne suffit pas si l'organisation ne sait pas précisément où se trouvent les produits et versions concernés. Le rapprochement futur entre alertes CVE et référentiel applicatif répond directement à cette difficulté.

La cartographie devient également déterminante lorsqu'il faut préparer la continuité ou la reprise du système d'information. L'ANSSI rappelle que la remédiation d'une cyberattaque suppose de prioriser les actifs, d'identifier leurs dépendances et de planifier l'ordre de reconstruction des services. Formaliser un PRA devient donc difficile lorsqu'une organisation ne connaît pas suffisamment ses applications critiques, les composants sur lesquels elles reposent et les dépendances nécessaires à leur redémarrage.

Un référentiel de ce type conserve néanmoins plusieurs limites. Sa qualité dépend d'abord des données qui l'alimentent : un équipement non supervisé ou une information absente de la source ne peut pas être reconstitué automatiquement. Certaines normalisations peuvent également nécessiter une validation humaine, tandis qu'un inventaire purement technique ne suffit pas à identifier systématiquement le rôle métier d'un logiciel, son responsable, sa criticité ou l'ensemble de ses dépendances.

Surtout, un référentiel peut constater la coexistence de nombreuses versions, mais il ne peut pas déterminer à lui seul lesquelles doivent être autorisées. L'outil décrit l'état réel ; l'organisation doit définir l'état attendu. Sa pérennité repose donc également sur des règles de gouvernance, des responsabilités clairement identifiées et des mécanismes permettant de maintenir cette connaissance dans le temps.

Cette distinction rejoint directement le sujet de ce mémoire. Une version supplémentaire conservée, une exception non documentée ou une application ajoutée sans être intégrée à la cartographie représente rarement un coût important au moment où la décision est prise. Pourtant, l'accumulation de ces écarts peut progressivement rendre le système d'information plus difficile à comprendre, à sécuriser et à faire évoluer.

Ce constat m'a conduit, dans les missions suivantes, à accorder davantage de place à la formalisation, aux règles et à la traçabilité, afin de rendre le système d'information non seulement plus lisible, mais aussi plus maîtrisable. Cette logique s'est notamment poursuivie avec la formalisation et l'automatisation du traitement des incidents de cybersécurité.

PRÉSENTATION DES TRAVAUX RÉALISÉS

03

3.3.1. CONTEXTE, MOTIVATIONS ET OBJECTIFS

Mieux connaître le système d'information permet de savoir ce qu'il faut protéger. Il faut néanmoins être en capacité de réagir suffisamment vite lorsque cette protection est contournée. Cette question prenait une importance particulière dans un contexte où la pression cyber sur les organisations publiques reste élevée. En 2025, l'ANSSI a eu connaissance de 1 366 incidents de sécurité en France, un niveau comparable à 2024, tandis que l'ENISA identifie l'administration publique comme le secteur le plus ciblé dans l'Union européenne, avec 38,2 % des incidents recensés sur la période étudiée.

Cette menace ne s'arrête pas aux protections techniques. L'utilisateur constitue lui-même un point de contact privilégié pour l'attaquant. Le Data Breach Investigations Report 2026 de Verizon estime ainsi que le facteur humain intervient dans 62 % des violations de données analysées. L'agent ne doit cependant pas être considéré uniquement comme une vulnérabilité. Correctement sensibilisé, il devient aussi l'un des premiers capteurs d'une attaque. Un courriel suspect identifié puis signalé rapidement peut permettre à la cellule cybersécurité de rechercher d'autres destinataires, d'évaluer l'étendue de la tentative et, si nécessaire, d'engager des mesures de protection avant qu'une compromission ne se propage. La sensibilisation constitue donc une première barrière ; encore faut-il que le signalement déclenche ensuite une réponse claire et efficace.

Cette problématique était directement observable au sein de la Région Guadeloupe. La cellule cybersécurité devait faire face chaque semaine à plusieurs tentatives ou signalements de sécurité, notamment liés au phishing et à la messagerie. Or, comme présenté précédemment, cette cellule reposait alors sur un seul agent permanent, auquel je venais apporter un appui dans le cadre de mon apprentissage. Dans ce contexte, la répétition d'opérations manuelles : rechercher un courriel, identifier ses destinataires, interroger les utilisateurs concernés, vérifier leurs réponses puis engager les contrôles nécessaires mobilisait un temps qui devait parallèlement rester disponible pour les autres activités de cybersécurité.

Le problème ne consistait donc pas uniquement à détecter davantage d'incidents, mais à transformer chaque signalement en une réponse suffisamment rapide, reproductible et traçable. Un agent devait savoir où signaler, quelles informations transmettre et quelle conduite adopter ; de son côté, la cellule devait pouvoir déterminer rapidement qui était concerné, quelles vérifications effectuer, quelles actions engager et comment conserver la trace du traitement. Sans cadre commun, chaque nouveau cas risquait de nécessiter de reconstruire une partie de cette chaîne de décision.

La mission poursuivait ainsi quatre objectifs complémentaires : sensibiliser et clarifier le signalement pour les agents, formaliser les responsabilités et les étapes de traitement, tendre vers une réponse semi-automatisée pilotée par la cellule cybersécurité, et sécuriser cette réponse grâce à des contrôles et une traçabilité suffisants.

3.3.2. FORMALISATION DES PROCÉDURES

Avant d'outiller le traitement des incidents, une première difficulté devait être résolue : rendre la cellule cybersécurité identifiable et son rôle compréhensible par les agents. Un canal spécifique a ainsi été créé afin de disposer d'un point d'entrée unique pour les signalements liés à la cybersécurité. Restait toutefois que les agents sachent dans quelles situations l'utiliser et quelles actions adopter lorsqu'ils étaient confrontés à un événement suspect.

Un premier travail de communication et de sensibilisation a donc été mené autour de procédures destinées aux utilisateurs. Elles précisaient notamment la conduite à tenir face à un courriel suspect : ne pas interagir avec son contenu, conserver les éléments utiles à l'analyse et transmettre le signalement à la cellule cybersécurité. La cible était de réduire les comportements susceptibles d'aggraver l'incident et de fournir suffisamment d'informations pour permettre une qualification rapide.

Afin que ce dispositif ne repose pas uniquement sur des communications ponctuelles, une note interne a également été élaborée. Elle officialisait l'existence de la cellule cybersécurité, son rôle ainsi que les principaux moyens permettant de la contacter, notamment son adresse électronique et son numéro de téléphone.

La formalisation devait cependant couvrir les deux côtés du signalement. Il était également nécessaire de définir ce que la cellule devait faire une fois l'alerte reçue. Les procédures internes ont donc précisé les vérifications à effectuer selon la nature et la gravité de l'incident, les informations à rechercher, les interlocuteurs à mobiliser ainsi que les éventuelles mesures correctives. Certaines pouvaient avoir un impact direct sur l'utilisateur ou son compte comme une réinitialisation des moyens d'authentification ou une révocation des sessions actives, ce qui imposait de préciser les responsabilités et les conditions de ces actions.

Ce premier cadre a ensuite servi de base à plusieurs procédures complémentaires. À la suite de la procédure de déclaration d'un incident, j'ai notamment pris l'initiative d'en produire une version plus visuelle, afin de rendre les consignes essentielles plus rapidement accessibles aux agents. La logique retenue devenait alors :
signaler → qualifier → rechercher l'étendue → décider → agir → suivre et tracer

Les procédures et les supports de communication étant confidentiels, seul le visuel est présenté en annexe E (figure E.1). Après avoir déterminé les actions à réaliser, leur ordre, par qui et leurs conditions d'exécution, cette procédure formalisée constituait désormais la base à partir de laquelle certaines étapes répétitives pouvaient être traduites en opérations automatisées.

3.3.3. CONCEPTION DE L'AUTOMATISATION POWERSHELL

La formalisation du processus permettait désormais de savoir quoi faire face à un signalement. Restait à déterminer ce qui pouvait réellement être automatisé. Cette mission présentait pour moi une difficulté particulière : elle reposait sur PowerShell, langage que je maîtrisais encore peu, tout en manipulant des opérations susceptibles d'avoir des conséquences directes sur les comptes et les messageries des agents. À cela s'ajoutait une forte dépendance à l'écosystème Microsoft : selon les étapes, le script devait interagir avec Exchange Online, Outlook, Microsoft Entra, SharePoint ou encore les services Microsoft 365 associés. Chaque nouvelle fonction nécessitait donc de comprendre non seulement sa logique, mais également les commandes disponibles, leurs prérequis, leurs permissions et leurs effets.

Dans ce contexte, développer directement sur des incidents réels aurait été difficilement acceptable. Des journaux d'exécution pouvaient vérifier qu'une fonction avait été appelée, mais pas toujours que son effet réel correspondait au comportement attendu. J'ai donc commencé par construire un environnement de test isolant explicitement les utilisateurs réels. Lorsque ce mode était activé, les destinataires, comptes et informations utilisés par les fonctions étaient remplacés par des valeurs de test définies à l'avance. Un courriel destiné théoriquement à une victime de phishing pouvait ainsi être réellement généré et envoyé, mais uniquement vers les comptes choisis pour les essais.

Cette séparation m'a permis de constituer une batterie de scénarios reproductibles visible sur le tableau 3.6, associant chaque situation à l'action attendue et au moyen permettant de la vérifier.

Tableau 3.6 — Exemples de scénarios reproductibles utilisés pendant le développement

Scénario simulé	Rôle joué	Comportement attendu	Vérification
Réception d'un phishing par un agent	Mon compte utilisé comme victime	Le message est retrouvé et l'utilisateur concerné identifié	Trace Exchange et résultat du script
Campagne touchant plusieurs agents	Comptes de test prédéfinis	Tous les destinataires concernés sont recensés sans contacter d'autres agents	Comparaison avec la population injectée
Demande de confirmation à une victime	Mon compte comme victime, mon tuteur comme destinataire de contrôle	Le modèle de courriel correspondant est envoyé avec l'identifiant de l'incident	Réception réelle du message
Réponse type	Réponse envoyée depuis le compte victime	L'incident est mis à jour et l'utilisateur rejoint la population	Fichier de suivi et journal
Réponse type	Réponse envoyée depuis le compte victime	La réponse est enregistrée sans déclencher l'escalade prévue pour ces réponses	Fichier de suivi

Le script a parallèlement été structuré afin de séparer l'orchestration des traitements de leurs fonctions techniques. Un lanceur principal pilotait le scénario à exécuter, tandis qu'un module regroupait les fonctions réutilisables : recherche d'un message et de ses destinataires, préparation des communications, lecture des réponses, mise à jour du suivi, contrôles complémentaires ou préparation des actions correctives. La représentation de cette architecture se trouve en annexe E (figure E.3).

La logique pouvait être représentée de manière simplifiée comme suit :
signalement → recherche → identification des personnes impactées → communication → surveillance → qualification → action → traçabilité

Cette architecture faisait apparaître un second enjeu : toutes les erreurs ne devaient pas avoir les mêmes conséquences. L'échec d'une opération critique devait pouvoir interrompre les traitements qui en dépendaient, tandis qu'une action secondaire en erreur ne devait pas nécessairement empêcher l'ensemble du scénario de continuer. Il fallait donc expliciter les dépendances, les priorités et les comportements attendus en cas d'échec.

Certains scénarios reproductibles ont alors été traduits en **tests Pester**, c'est-à-dire des tests automatisés pour PowerShell, afin de vérifier ponctuellement le bon fonctionnement du script et de formaliser les actions qu'il était autorisé à exécuter dans chacun des deux environnements.

Voici un exemple de scénario pouvant imposer les règles suivantes dans le tableau 3.7 ci-dessous :

Tableau 3.7 — Comportements attendus selon le mode d'exécution de CyberAlerte

Situation	Mode test	Mode réel
Destinataire issu de l'incident	Remplacé par le compte de test autorisé	Utilisateur réellement identifié
Échec de la recherche du message	Arrêt des étapes dépendantes	Même comportement attendu
Échec d'un envoi secondaire	Journalisation puis poursuite des traitements indépendants	Même comportement attendu
Action sensible non explicitement autorisée	Refus / simulation	Soumise aux contrôles prévus avant exécution

Les tests permettaient ainsi de vérifier non seulement les résultats positifs, mais également une propriété essentielle : le script devait échouer de la manière prévue. Une évolution du code ne devait pas, par exemple, transformer une opération simulée en action réelle ou permettre à une étape dépendante de continuer alors que son prérequis avait échoué.

Le processus a ensuite évolué à mesure que les premiers scénarios devenaient opérationnels. Selon la procédure définie, lorsqu'une tentative de phishing avait atteint un ou plusieurs agents, le script pouvait leur adresser un message adapté. Certains types de réponse ou l'absence de réponse pouvaient justifier des investigations ou mesures supplémentaires. Une vérification ponctuelle immédiatement après l'envoi n'était donc pas suffisante.

Une surveillance sur plusieurs heures a été ajoutée au scénario. Après l'envoi des messages, le script devait conserver la liste des personnes concernées, interroger périodiquement les réponses reçues, les rattacher à l'incident correspondant et actualiser leur état. À l'issue de la fenêtre de surveillance, trois situations pouvaient ainsi être distinguées :

- traitement complémentaire
- information enregistrée
- situation signalée pour suivi

Cette évolution a nécessité d'introduire une véritable gestion d'état : le traitement ne pouvait plus être considéré comme une simple suite de commandes exécutée une fois, puisqu'il devait retrouver plusieurs heures plus tard les personnes déjà contactées, les réponses déjà traitées et les actions restantes à effectuer.

Cette question a conduit à faire évoluer la traçabilité du script. Mon maître d'apprentissage avait initialement proposé de conserver une liste des agents en fonction des réponses. J'ai prolongé cette idée en structurant un fichier maître, servant de référence pour chaque incident, puis des vues dédiées permettant de suivre séparément les personnes concernées et les opérations réalisées.

La logique retenue reposait notamment sur :
- maitre.csv, regroupant les informations principales permettant d'identifier et de suivre l'incident ;
- suivi.csv, conservant l'état des personnes concernées et leurs éventuelles réponses ;
- actions.csv, enregistrant les actions associées au traitement, leur date et les informations nécessaires à leur traçabilité.

Il devenait ainsi possible de reconstituer non seulement ce qui avait été détecté, mais également qui avait été contacté, quelle réponse avait été obtenue et quelles opérations le script avait réalisées ou préparées.

L'automatisation commençait à transformer la procédure en un processus testable, suivi dans le temps et capable de conserver son propre état.

Cette conception faisait néanmoins apparaître une dernière frontière entre ce qui pouvait être programmé et ce qui pouvait réellement être exécuté. Certaines fonctions fonctionnaient correctement dans l'environnement de test mais nécessitaient, dans l'environnement de la collectivité, des habilitations Microsoft supplémentaires ou concernaient des opérations suffisamment sensibles pour imposer des protections renforcées. La mise en œuvre réelle du script a donc nécessité d'examiner précisément les droits, les mécanismes de journalisation et les conditions dans lesquelles une action devait être autorisée, simulée ou bloquée.

3.3.4. MISE EN ŒUVRE ET SÉCURISATION

Le passage des scénarios de test à l'environnement réel a rapidement fait apparaître une contrainte supplémentaire : une fonction techniquement développée ne pouvait être réellement exécutée que si le compte utilisé disposait des **habilitations nécessaires** sur les différents services Microsoft mobilisés.

Les principales fonctions de recherche, d'identification des utilisateurs concernés, de génération des communications, de suivi et de traçabilité ont ainsi pu être validées dans l'environnement de test. En revanche, plusieurs fonctionnalités n'ont pas pu être éprouvées intégralement en conditions réelles faute de droits suffisants. C'était notamment le cas de certaines actions sensibles sur les comptes.

Cette limitation a renforcé le choix effectué dès la conception : séparer ce que le script sait préparer de ce qu'il est effectivement autorisé à exécuter. Les opérations susceptibles d'avoir un impact important sur un utilisateur ou son environnement devaient rester bloquées ou simulées lorsque les conditions d'autorisation n'étaient pas réunies. Une erreur de permission ne devait par ailleurs jamais être interprétée comme une opération réussie ni permettre la poursuite d'une étape qui en dépendait.

Pour chaque solution développée, je définis **dès sa conception** la manière dont elle devra être **évaluée au fil de son développement puis avant sa livraison**. Cette démarche permet de ne pas attendre la fin du projet pour vérifier sa qualité, mais de disposer dès le départ de critères permettant de contrôler progressivement son fonctionnement, sa fiabilité et ses limites. Elle me paraît particulièrement importante lorsqu'une solution doit être remise à une organisation, qui doit pouvoir disposer d'éléments objectifs sur ce qu'elle peut réellement en attendre. Dans le cas de CyberAlerte, cette réflexion a été formalisée dans une matrice d'évaluation présentée en figure 3.10, associant chaque dimension critique de la solution aux méthodes retenues pour la vérifier.

Figure 3.10 — Matrice des méthodes d'évaluation pour CyberAlerte

Solutions de tests CYBER		
Validation retenue	Scénarios reproductibles	Vérifier le comportement attendu sur les cas nominaux et limites
	Tests Pester	Garantir les comportements critiques et limiter les régressions
	Injection de défaillances	Vérifier qu'une erreur ne provoque pas d'action dangereuse ou incohérente
	PSScriptAnalyzer	Identifier les mauvaises pratiques et anomalies potentielles du code
	Manuel vs automatisé	Mesurer ensuite le gain de temps et la réduction des interventions humaines

Cette stratégie reprend les axes d'évaluation jugés les plus pertinents pour un outil de réponse à incident : efficacité opérationnelle, fiabilité des scénarios et robustesse face aux défaillances, la qualité du code venant compléter ces preuves.

Pour des raisons de confidentialité et de sécurité, aucune capture des interfaces d'administration, des comptes ou des informations manipulées ne sera reproduite dans le mémoire.

Les limitations d'habilitation empêchent donc de considérer l'ensemble du workflow comme validé de bout en bout en production. Elles n'empêchent toutefois pas d'évaluer les fonctions effectivement testables et, surtout, de mesurer si l'automatisation apporte un gain réel par rapport au traitement manuel. C'est sur ces deux dimensions, fiabilité et efficacité opérationnelle, que porte l'évaluation présentée dans la section suivante.

3.3.5. RÉSULTATS, APPORTS ET LIMITES

Les expérimentations réalisées ont permis de valider les principales briques de CyberAlerte : recherche des éléments associés à un incident, identification et suivi des agents concernés, génération des communications, prise en compte des réponses et conservation d'une trace des traitements effectués. Les scénarios reproductibles et les tests Pester ont également permis de vérifier que plusieurs situations dégradées conduisaient au comportement attendu, notamment l'arrêt d'une action dépendante lorsqu'un prérequis échouait ou le blocage d'une opération sensible hors du périmètre autorisé.

L'apport principal du script réside davantage dans la transformation d'une succession d'actions manuelles en un processus reproductible et traçable. Une partie des recherches, communications, contrôles et mises à jour du suivi peut être préparée ou exécutée automatiquement, réduisant le nombre d'opérations répétitives laissées à la cellule cybersécurité. La conservation d'un historique des incidents, des agents concernés et des actions réalisées permet également de ne plus dépendre uniquement de la mémoire de l'opérateur pour reconstituer le traitement effectué.

Tableau 3.8 — Synthèse de l'évaluation de CyberAlerte

Dimension évaluée	Protocole	Résultat principal
Efficacité opérationnelle	Comparaison manuel / CyberAlerte sur cinq scénarios	−86 % de temps moyen et −77 % d'interventions humaines
Fiabilité fonctionnelle	80 exécutions réparties sur huit scénarios reproductibles	78/80 réussites — 97,5 %
Robustesse	40 défaillances injectées volontairement	38/40 correctement gérées — 95 % ; aucune action dangereuse déclenchée

Sur les scénarios mesurés, le temps moyen est ainsi passé de 7 min 53 s à 1 min 05 s, soit une diminution d'environ 86 %. Le nombre moyen d'interventions humaines passait parallèlement de 9,6 à 2,2, soit une réduction d'environ 77 %. L'intérêt de l'automatisation ne résidait donc pas uniquement dans sa vitesse : elle retirait surtout à l'opérateur une partie des recherches, manipulations et opérations de traçabilité répétitives.

Les tests de robustesse allaient dans le même sens. Sur 40 défaillances provoquées volontairement, 38 ont produit le comportement prévu, soit 95 %. Surtout, aucune action dangereuse n'a été déclenchée lors de ces erreurs. Pour un outil de réponse à incident, ce résultat est particulièrement important : face à une donnée incomplète, un problème d'authentification ou une indisponibilité de service, il est préférable que l'automatisation interrompe ou signale le traitement plutôt qu'elle poursuive à partir d'une hypothèse incertaine.

Ces résultats restent néanmoins ceux d'un prototype partiellement validé. Certaines fonctions n'ont pas pu être éprouvées de bout en bout dans l'environnement réel faute d'habilitations suffisantes. Les résultats associés à ces fonctionnalités doivent donc être distingués des fonctions effectivement validées en conditions réelles. Le stockage dans maitre.csv, suivi.csv et actions.csv constitue également une solution adaptée à l'expérimentation, mais devrait évoluer vers un stockage plus structuré en cas d'industrialisation.

Enfin, CyberAlerte ne doit pas être considéré comme un outil capable de décider seul de la réponse à apporter à un incident. Toute action reste déclenchée ou validée par une intervention humaine.

Cette dernière limite soulevait finalement une question dépassant le seul traitement des incidents : pour automatiser davantage sans augmenter le risque, il fallait précisément déterminer qui peut effectuer quelle action, sur quelle ressource et avec quel niveau de privilège. Cette problématique se retrouvait directement dans le chantier suivant consacré à la gouvernance des identités et des accès.

PRÉSENTATION DES TRAVAUX RÉALISÉS

03

3.4.1. CONTEXTE, RISQUES ET OBJECTIFS

Les difficultés rencontrées lors de l'automatisation du traitement des incidents avaient déjà fait apparaître une question essentielle : disposer d'un outil capable d'effectuer une action ne signifie pas que n'importe quel acteur doit pouvoir l'exécuter. Cette interrogation dépassait largement CyberAlerte et concernait plus généralement la manière dont les identités et les habilitations étaient gérées dans le système d'information.

Au sein de la collectivité, de nombreuses applications utilisent Active Directory pour gérer les comptes et les accès dans le réseau interne, tandis que Microsoft Entra prend progressivement une place plus importante pour les applications et ressources cloud. La coexistence de ces deux environnements conduit ainsi à une gestion hybride des identités et des accès. Certaines applications restent toutefois gérées directement par leurs responsables fonctionnels, sans pilotage centralisé complet par la DSI.

Cette organisation pose notamment problème lors du départ d'un agent : désactiver son compte principal ne suffit pas toujours à garantir que tous ses accès aux différentes applications ont bien été supprimés.

Le problème ne résidait donc pas seulement dans la création ou la suppression technique d'un compte. Il concernait le cycle de vie[16] de l'identité : que se passe-t-il lorsqu'un agent arrive, change de fonction, rejoint une autre direction, cumule temporairement plusieurs responsabilités ou quitte la collectivité ? Microsoft structure lui-même cette problématique autour du modèle Joiner–Mover–Leaver, dans lequel l'identité et ses accès doivent évoluer avec les différentes phases du parcours de l'utilisateur.

Un droit accordé à un instant donné peut être parfaitement légitime, puis devenir inadapté plusieurs mois plus tard si la situation de l'agent évolue. Le principe du moindre privilège consiste précisément à limiter les autorisations aux ressources strictement nécessaires à l'accomplissement des fonctions confiées. Mais appliquer ce principe suppose de pouvoir répondre à des questions simples en apparence : qui possède quel droit, pourquoi, qui l'a validé, depuis quand et jusqu'à quand ?

Le chantier IAM[17] (Identity and Access Management) poursuivait donc plusieurs objectifs : centraliser la connaissance des identités et habilitations, rendre les validations traçables, sécuriser les mouvements et départs, permettre des revues périodiques et préparer progressivement l'automatisation du cycle de vie. Le cadrage réalisé ajoutait une exigence forte : conserver un historique non destructif permettant de reconstituer l'état des comptes et des droits à une date donnée.

3.4.2. FORMALISATION DU CYCLE DE VIE ET DES EXIGENCES DE GOUVERNANCE

Avant d'étudier une solution IAM, il fallait d'abord comprendre ce que celle-ci aurait à gouverner. J'ai donc commencé par formaliser les principaux événements du cycle de vie d'un agent et les interactions nécessaires entre les différents acteurs.

La représentation BPMN (Business Process Model and Notation), une notation graphique standardisée permettant de représenter les étapes, acteurs et enchaînements d'un processus métier, présentée en annexe F (figure F.1), permettait notamment de distinguer le déclenchement métier d'une demande, sa validation et son exécution technique.

Cette représentation faisait apparaître que l'administrateur technique ne pouvait pas être le seul responsable du processus. Le responsable métier est généralement mieux placé pour déterminer si le droit est justifié, tandis que l'équipe technique garantit son application dans le système cible. De la même manière, un départ ou une mobilité nécessite que l'information parvienne suffisamment tôt aux acteurs chargés de modifier les accès.

L'analyse du processus a ainsi permis de transformer un besoin relativement général d'« IAM » en exigences plus précises, comme le démontre le tableau 3.9 suivant.

16. Ensemble des étapes successives par lesquelles passe un élément, de sa création ou son arrivée jusqu'à sa suppression ou sa fin d'utilisation

17. Ensemble de règles, processus et outils permettant de gérer les identités numériques et de contrôler leurs droits d'accès aux ressources d'un système d'information

Tableau 3.9 — Exigences de gouvernance relevées

Exigence	Question à laquelle le dispositif doit répondre
Cycle de vie	Les accès suivent-ils correctement les arrivées, mobilités et départs ?
Responsabilités	Qui demande, qui valide et qui exécute ?
Traçabilité	Peut-on expliquer qui a accordé quoi, quand et pourquoi ?
Historisation	Peut-on reconstituer les droits détenus à une date passée ?
Revue des accès	Les droits encore présents restent-ils justifiés ?
Réconciliation	Les droits attendus correspondent-ils à ceux réellement présents ?
Séparation des rôles	Une même personne peut-elle demander, valider et appliquer un droit sensible sans contrôle ?
Reporting	Les comptes à risque, droits sensibles et écarts sont-ils visibles ?

Ces exigences correspondaient directement aux besoins qui ont ensuite structuré le travail de sourcing : cycle Joiner–Mover–Leaver, révocation lors des départs, comptes orphelins, historisation, demandes d'accès, séparation des tâches, certification périodique, réconciliation[18] ou encore provisioning[19] et déprovisioning[20]. Le processus complet figure à l'annexe F (figure F.1).

L'enjeu se déplaçait ainsi progressivement : il ne s'agissait plus seulement de gérer des comptes, mais de **gouverner** la relation entre une personne, sa fonction, les droits dont elle dispose et les décisions ayant conduit à leur attribution.

18. Processus consistant à comparer plusieurs sources de données afin d'identifier les écarts et de rétablir leur cohérence.
19. Processus consistant à créer, configurer et attribuer automatiquement les ressources ou droits nécessaires à un utilisateur dans un système d'information.
20. Processus consistant à supprimer ou retirer les comptes, ressources et droits d'accès d'un utilisateur lorsqu'ils ne sont plus nécessaires

3.4.3. D'UNE RÉPONSE TRANSITOIRE À L'ÉTUDE D'UNE SOLUTION IAM

La mise en place immédiate d'une plateforme IAM complète n'était ni nécessairement possible ni souhaitable avant d'avoir suffisamment clarifié les processus et les données à gouverner. Une réponse transitoire a donc été utilisée sous la forme d'une liste SharePoint dédiée au suivi des habilitations. Celle-ci constituait déjà une réalisation opérationnelle utilisée pour améliorer la visibilité et la traçabilité des droits.

Cette solution apportait une première centralisation là où l'information pouvait auparavant être répartie entre plusieurs acteurs. Elle permettait surtout de commencer à structurer la connaissance avant de chercher à l'automatiser.

Ses limites apparaissaient cependant rapidement. Une liste SharePoint peut conserver et présenter l'information, mais elle ne constitue pas à elle seule un véritable système de gouvernance des identités : la qualité du suivi reste dépendante des saisies humaines, les droits effectivement présents dans les applications ne sont pas automatiquement réconciliés avec ceux enregistrés et les arrivées, mobilités ou départs ne déclenchent pas nécessairement une modification automatique des systèmes concernés.

La liste devait donc être considérée comme une étape intermédiaire et non comme la cible.

Objectiver le choix d'une solution IAM/IGA

Une démarche de sourcing a parallèlement été engagée afin d'étudier des solutions IAM/IGA[21] (Identity Governance and Administration) du marché. Le travail réalisé ne s'est pas limité à demander une présentation commerciale. Un questionnaire commun de 117 questions, dont 76 identifiées comme essentielles, a été construit et je me suis notamment appuyé sur le **Guide de l'achat public – Le sourcing opérationnel**, publié par la Direction des achats de l'État. L'édition initialement utilisée, publiée en 2019, propose une démarche organisée autour de l'identification des acteurs, de l'organisation des échanges avec les fournisseurs et de l'exploitation des résultats. Le guide fournit également des documents types conçus pour être adaptés aux différents projets d'achat pour examiner notamment la couverture fonctionnelle, les intégrations, l'architecture, la sécurité, la mise en œuvre, l'exploitation, les coûts et la réversibilité. Le même questionnaire, la même durée d'entretien et le même niveau d'information ont été prévus pour chaque fournisseur afin de rendre la démarche comparable et traçable. Le but : produire un modèle suffisamment générique pour pouvoir être repris lors de futures études de solutions, tout en laissant les critères spécifiques au projet être adaptés.

Le cadrage demandait notamment aux prestataires de couvrir l'intégration à Active Directory, Entra ID, au SIRH, aux applications métiers, à l'ITSM[22] (IT Service Management) et aux outils de supervision, ainsi que la possibilité de reconstruire historiquement les droits et de gérer leur cycle de vie.

J'ai pour habitude de créer une matrice d'évaluation utilisant des critères pondérés pour la comparaison de plusieurs solutions. Elle est composée de 6 étapes : le contexte et les objectifs, le périmètre retenu (quelles solutions à comparer), la méthodologie et les critères d'évaluation, la matrice avec critères pondérés, un comparatif de coûts estimatif (avec un échantillon de 1500 personnes en général), et le SWOT[23] (Strengths, Weaknesses, Opportunities, Threats) des solutions avec les meilleures notes et par solutions.

Chaque solution peut être notée sur une échelle commune, par exemple de 0 à 5, puis pondérée selon l'importance du critère. Une note élevée ne doit cependant être attribuée que lorsque l'affirmation du fournisseur est accompagnée d'une démonstration, d'une documentation ou d'un élément vérifiable. Cette discipline était d'ailleurs explicitement prévue dans le fichier de sourcing : une information publique ou commerciale devait rester « à confirmer » tant qu'une preuve n'avait pas été apportée.

Cette approche permet de distinguer deux niveaux complémentaires :
questionnaire détaillé → comprendre et vérifier la solution
matrice pondérée → comparer et éclairer la décision

La matrice complète, les pondérations et les résultats anonymisés sont présentés en annexe F (figures F.2 à F.4).

21. Ensemble de processus et d'outils permettant de gouverner les identités, les droits d'accès et leur cycle de vie au sein d'un système d'information

22. Ensemble de pratiques permettant d'organiser, fournir, gérer et améliorer les services informatiques d'une organisation

23. Méthode d'analyse stratégique permettant d'identifier les forces, faiblesses, opportunités et menaces liées à un projet, une organisation ou une solution.

3.4.4. VALIDATION, APPORTS ET LIMITES

La mission a ainsi produit **trois apports complémentaires** : un support opérationnel immédiatement utilisable avec SharePoint, une formalisation du cycle de vie et des responsabilités sous forme de processus, et un cadre de sourcing[24] réutilisable pour objectiver de futures décisions d'achat. Le processus formalisé permettait alors de déduire les effets attendus des principaux mouvements d'un agent, synthétisés dans le tableau 3.10.

Tableau 3.10 — Mouvements des agents et processus de droits et rôles associés

Mouvement de l'agent	Effet attendu sur les droits	Actions principales
Arrivée	Création des accès nécessaires au poste	Création du compte, attribution des groupes/rôles selon le métier
Mobilité interne	Adapter les accès au nouveau poste	Retirer les anciens droits, attribuer les nouveaux
Changement temporaire de fonction	Accès supplémentaires limités dans le temps	Attribution avec date d'expiration et validation
Départ	Suppression de la capacité d'accès	Désactivation du compte, révocation des sessions et droits
Retour / réintégration	Réévaluer les droits nécessaires	Réactivation contrôlée, nouvelle validation

Cette grille fait apparaître la principale contribution de la réponse transitoire : **améliorer la visibilité et conserver une trace plus structurée des habilitations et des décisions associées**. Elle permet également de mieux préparer une future solution IAM en clarifiant les informations qu'il faudra importer, les acteurs qu'il faudra intégrer aux workflows[25] et les événements qui devront déclencher des actions.

Elle ne résout toutefois pas encore le problème dans son ensemble. La liste SharePoint reste dépendante d'une mise à jour humaine et ne garantit pas que l'état enregistré corresponde à l'état réel des droits dans chaque application. Une solution IAM durable devra notamment être capable de réconcilier ces deux réalités, d'automatiser progressivement le provisioning et le déprovisioning et de produire la preuve que les actions attendues ont effectivement été exécutées. Ces fonctions constituent précisément plusieurs des exigences majeures du questionnaire de sourcing.

La technologie ne constitue cependant qu'une partie de la réponse. Même une plateforme IAM complète ne peut déterminer seule qui doit valider un accès, quel responsable métier en est propriétaire, quelle source fait autorité ou à quelle fréquence un droit doit être revu. Ces décisions doivent exister avant de pouvoir être automatisées.

La question de la conservation et de la transmission de l'information ne se limitait d'ailleurs pas aux habilitations. Elle se retrouvait également dans la documentation et les connaissances produites par l'organisation, mais dispersées, ce qui a conduit à explorer un dernier chantier : **l'utilisation de l'intelligence artificielle (IA) pour faciliter l'accès à la connaissance documentaire**.

24. Document définissant les critères, étapes et bonnes pratiques à suivre pour identifier, comparer et sélectionner des fournisseurs ou des solutions adaptés à un besoin.

25. Enchaînement structuré d'étapes ou d'actions permettant de réaliser une tâche ou un processus

PRÉSENTATION DES TRAVAUX RÉALISÉS

03

3.5.1. DU CAS ACADÉMIQUE À UNE OPPORTUNITÉ POUR LA COLLECTIVITÉ

Ce dernier chantier suivait une trajectoire différente : **il est né de ma propre initiative**, à partir d'un travail académique que j'ai choisi de transposer à mon environnement professionnel.

L'intérêt de cette transposition est apparu progressivement au regard des difficultés rencontrées pendant l'apprentissage. Le projet CKAN avait montré les conséquences que pouvait avoir une information technique insuffisamment documentée lorsqu'elle restait principalement détenue par quelques personnes. La cartographie du système d'information avait nécessité de rapprocher des informations dispersées entre outils, fichiers et interlocuteurs. La formalisation du traitement des incidents avait elle aussi consisté à transformer des connaissances et habitudes de travail en procédures explicites. Enfin, le chantier IAM avait montré combien il était difficile de gouverner une information lorsqu'elle était répartie entre plusieurs acteurs et systèmes.

Ces situations étaient différentes, mais faisaient apparaître un problème commun : une organisation peut disposer de beaucoup d'informations sans pour autant **disposer facilement de la connaissance au moment où elle en a besoin**.

Une procédure peut exister sans être connue de l'agent qui en a besoin. Un rapport peut contenir la réponse recherchée sans que son emplacement soit évident. Une décision ancienne peut être documentée mais difficile à retrouver. Plusieurs documents peuvent enfin devoir être rapprochés pour comprendre une situation qui, elle, dépasse leurs frontières respectives.

Cette problématique dépasse d'ailleurs le seul contexte de la collectivité. Dans son Work Trend Index 2023, Microsoft indiquait que 62 % des personnes interrogées estimaient consacrer trop de temps à rechercher de l'information dans leur journée de travail. Le problème n'est donc pas uniquement de produire davantage de documentation, mais aussi d'en rendre l'exploitation suffisamment rapide.
C'est dans ce contexte que j'ai identifié une possibilité de transposer un projet académique consacré à l'intelligence artificielle vers un assistant documentaire destiné à interroger la connaissance interne de la collectivité.

L'assistant devait permettre à un agent de poser directement une question telle que :
« **Quelle est la procédure à suivre lorsqu'un agent signale un courriel de phishing ?** » ou :
« **Quels documents décrivent l'architecture et les dépendances de cette application ?** » ou encore :
« **Quels incidents de cette catégorie ont été recensés au cours des derniers mois ?** »
Pour ce dernier usage, le projet envisageait également d'exploiter les métadonnées SharePoint et certaines informations contenues dans les documents afin de produire des **indicateurs** : nombre d'incidents par période, catégories rencontrées, gravité, services concernés ou tendances récurrentes. Le système pouvait ainsi devenir non seulement un outil de recherche, mais également un **support d'exploitation de la connaissance accumulée.**

Ma proposition visait donc plusieurs besoins complémentaires :
retrouver → comprendre → rapprocher → contextualiser → transmettre → exploiter

L'enjeu de transmission était particulièrement important. Une connaissance uniquement accessible parce qu'un agent sait « où chercher » reste **fragile**. En permettant d'interroger plusieurs sources à travers une interface commune et en renvoyant vers les documents ayant servi à produire la réponse, l'assistant pouvait contribuer à réduire cette dépendance.

Il ne supprimait toutefois pas la dette documentaire elle-même. Une information jamais documentée ne peut pas être retrouvée par une intelligence artificielle, et une source erronée ou obsolète peut conduire à une réponse tout aussi erronée. Le projet devait donc être considéré comme une couche facilitant l'accès à la mémoire organisationnelle, et non comme un substitut à sa constitution et à son maintien.

Mon idée a d'abord été soumise à mon maître d'apprentissage, qui en a validé l'intérêt et m'a autorisé à approfondir la proposition. Elle a ensuite été présentée au DSI, qui en a également validé l'intérêt pour la collectivité. Ce double accord a permis de faire passer une réflexion initialement académique au stade d'une véritable opportunité étudiée pour la collectivité.

3.5.2. CONCEVOIR, DOCUMENTER ET DÉFENDRE LA PROPOSITION

Ma proposition reposait sur un principe simple : l'assistant ne devait pas répondre uniquement à partir des connaissances générales d'un modèle de langage. Il devait d'abord rechercher les ressources internes pertinentes, transmettre leur contenu au modèle puis construire une réponse permettant à l'utilisateur de retrouver ses sources.

L'architecture envisagée suivait ainsi une logique de Retrieval-Augmented Generation (RAG), c'est-à-dire une génération de réponse enrichie par la recherche d'informations dans une base documentaire :

documents → extraction → indexation des contenus → recherche → sélection du contexte pertinent → LLM (grand modèle de langage) → réponse sourcée

Ce principe permet de construire des applications d'IA générative s'appuyant sur des connaissances documentaires retrouvées au moment de la requête, plutôt que sur les seules connaissances acquises par le modèle lors de son entraînement.

Les premières orientations prévoyaient la récupération de documents depuis SharePoint, leur préparation et leur indexation, puis la possibilité de les rechercher à partir d'une question formulée en langage naturel. Microsoft propose aujourd'hui plusieurs mécanismes permettant précisément d'utiliser des contenus SharePoint dans des architectures de recherche et de RAG.

La question des habilitations devait cependant être traitée dès la conception. Centraliser l'accès à la connaissance ne devait jamais signifier contourner les permissions existantes : un utilisateur ne devait pouvoir obtenir via l'assistant que les informations auxquelles il était lui-même autorisé à accéder. Les architectures modernes de recherche documentaire prévoient justement des mécanismes de contrôle d'accès au niveau des documents.

Mais concevoir une solution techniquement cohérente ne suffisait pas. Une initiative n'a de valeur dans une organisation que si elle peut être **comprise, discutée et défendue** auprès de ceux qui devront décider de son intérêt, l'utiliser ou permettre sa mise en œuvre.

J'ai donc préparé une présentation qui ne reposait pas uniquement sur la description de mon projet. Pour éviter de venir devant la direction avec une simple idée abstraite, j'ai cherché à construire une démonstration argumentée du besoin : **problèmes rencontrés** dans le fonctionnement quotidien, **cas d'usage** concrets, **architecture envisagée**, **valeur attendue** et **faits statistiques** permettant de replacer la problématique de recherche d'information dans un contexte plus large. Les cas d'usage présentés dans le tableau 3.11 permettent notamment de montrer ce que l'outil changerait concrètement :

Tableau 3.11 — Exemple de cas d'usage avec et sans la solution

Situation	Sans assistant	Usage envisagé
Retrouver une procédure	Rechercher son emplacement ou solliciter un collègue	Poser directement la question et accéder au document source
Comprendre un sujet technique	Lire plusieurs documents et identifier leurs relations	Obtenir une synthèse contextualisée renvoyant vers les sources
Reprendre un dossier ancien	Identifier les personnes connaissant son historique	Interroger les rapports et documents conservés
Analyser des incidents	Exploiter manuellement plusieurs documents ou tableaux	Extraire les informations et produire des indicateurs structurés
Arrivée d'un nouvel agent	Dépendre fortement de la transmission orale	Faciliter l'accès autonome aux procédures et connaissances existantes

Cette préparation m'a permis de présenter le projet devant environ 24 membres de la Direction de la Transition Numérique, en présence notamment du DSI et de la vice-directrice adjointe. **La proposition a reçu un accueil favorable de l'ensemble des participants**, qui se sont montrés satisfaits du travail présenté et de l'intérêt des cas d'usage envisagés.

Cette étape a constitué l'un des apprentissages professionnels importants de la mission. Présenter une idée devant une équipe entière, composée de profils ayant des responsabilités et des expertises différentes, oblige à sortir d'un raisonnement purement technique. Il faut être capable d'expliquer quel problème est résolu avant d'expliquer comment la technologie fonctionne, anticiper les objections, rendre les bénéfices concrets et être en mesure d'étayer ce que l'on avance.

J'en ai surtout retenu **qu'avoir une idée ne suffit pas pour la faire exister** dans une organisation. La proposer suppose de la préparer et de fournir suffisamment d'éléments pour permettre aux autres de l'évaluer : cas d'usage, faits, contraintes, architecture, risques et conditions de réussite.
Cette trajectoire constitue elle-même un résultat de l'apprentissage : elle m'a confronté à la capacité attendue d'un professionnel de ne pas seulement produire une solution, mais aussi de faire comprendre pourquoi elle mérite d'être étudiée. Des extraits de cette présentation se retrouvent en annexe G (figure G.1).

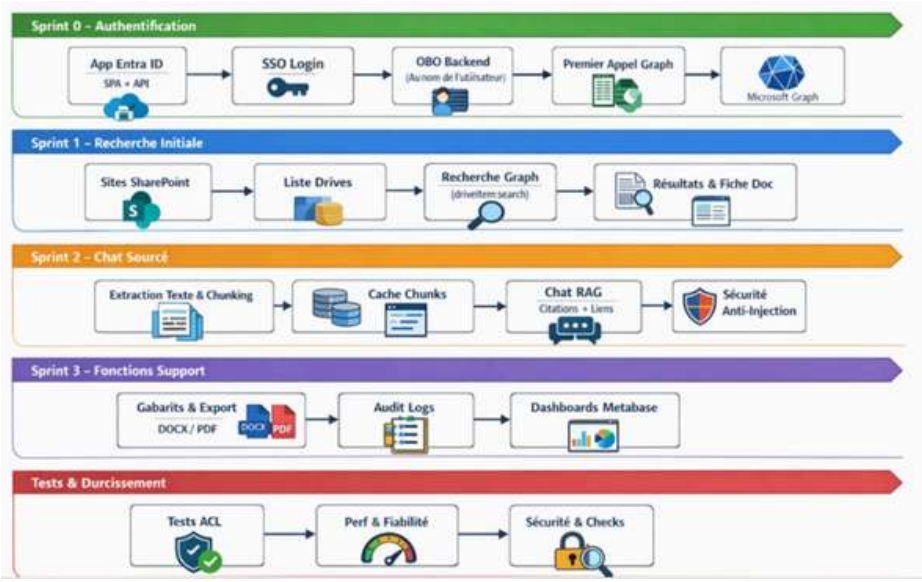
3.5.3. ARCHITECTURE ENVISAGÉE ET ARTEFACTS PRODUITS

Le projet n'ayant pas atteint le stade de l'implémentation, les travaux ont néanmoins conduit à la production de deux artefacts directement réutilisables.

Le premier est la présentation de la proposition, utilisée successivement pour exposer le problème, les cas d'usage, la valeur attendue et les orientations techniques au maître d'apprentissage, au DSI puis à l'ensemble de l'équipe.

Le second est un rapport de cadrage détaillé faisant office de cahier des charges initial. Il formalise les besoins identifiés, les fonctionnalités envisagées, les sources documentaires, l'architecture, les problématiques d'indexation, les exigences de sécurité et les conditions nécessaires à une éventuelle reprise.

Figure 3.11 — Architecture fonctionnelle envisagée de l'assistant documentaire



La chaîne technique pouvait être synthétisée ainsi :

- SharePoint / sources documentaires
  - collecte
  - extraction et préparation
  - indexation / vectorisation
  - recherche
  - sélection du contexte
  - modèle de langage
  - réponse accompagnée de ses références

Les usages étudiés ne se limitaient pas à la question-réponse. L'architecture devait également permettre d'envisager la synthèse de documents, leur rapprochement, l'extraction d'informations structurées et la production de certains indicateurs à partir des connaissances accumulées.

La feuille de route prévoyait une durée d'environ deux mois pour la réalisation d'un MVP, comme l'illustre la figure 3.12.

Figure 3.12 — Roadmap de l'assistant documentaire issue de la présentation

ROADMAP

Sprint	Période (jours ouvrés)	Objectif	Livrables clés
Sprint 0	10/03 → 23/03	Socle accès & cadrage	SSO Entra + **OBO Graph** (accès au nom de l'utilisateur), découverte sites/drives, squelette UI (Login / Documents), backlog + critères d'acceptation
Sprint 1	24/03 → 06/04	Recherche MVP	Recherche **Graph driveItem:search**, liste résultats + lien SharePoint, filtres simples (site/type/date), stockage métadonnées en Postgres
Sprint 2	07/04 → 20/04	Chat "sourcé" sur 1 document	Chat sur doc sélectionné, **2–5 citations + liens**, règle **"Je ne sais pas sans preuve"**, premiers logs (requêtes, doc IDs cités)
Sprint 3	21/04 → 04/05	Scan + Paramètres + Stats MVP	Page Paramètres (sources/exclusions), **scan initial + delta (état)**, Metabase **lien** + dashboards stock/usage (MVP), durcissement "anti prompt injection" (basique)
Sprint 4	05/05 → 18/05	Tâches + Gouvernance basique + Rôles	Templates → génération → édition → **export DOCX/PDF** + lien SP, rôles (Utilisateur/Auditeur/Admin), gouvernance basique (doublons simples/obsolescence heuristique), audit rétention (cadrage)
Jalon MVP	18/05	**MVP prêt pilote**	Démo complète "Recherche + Chat sourcé + Stats + Tâches + Paramètres + Gouvernance basique" sur périmètre 2 sites

3.5.4. BLOCAGE DU PROJET, LIMITES ET CONDITIONS DE REPRISE

La mise en œuvre nécessitait un environnement disposant des ressources permettant d'héberger les composants d'indexation, de recherche, les expérimentations associées et surtout le modèle IA choisi. L'accès au serveur et à des ressources complémentaires constituait donc un prérequis.

Contrairement au projet CKAN, le blocage rencontré ici ne provenait pas de la dépendance envers une personne. Le système d'information était alors engagé dans une migration visant à externaliser la gestion du serveur concerné auprès d'un prestataire. Durant cette transition, les nouveaux projets nécessitant des modifications sur cette infrastructure étaient suspendus afin de ne pas interférer avec la migration.

Son calendrier dépassant celui de la mission, les conditions permettant de réaliser correctement le prototype n'étaient plus réunies. Le projet a donc été interrompu avant cette phase plutôt que de construire une démonstration sur une infrastructure provisoire destinée à être modifiée.

La principale limite est par conséquent très claire : la proposition avait été validée et ses orientations jugées pertinentes, mais le contexte d'infrastructure n'a pas permis d'engager la phase de prototypage et donc d'évaluer la solution en conditions réelles.
Une reprise nécessiterait en outre davantage qu'un serveur disponible. Elle supposerait de préciser la gouvernance documentaire, la qualité et l'actualisation des sources, les règles d'accès, la confidentialité, le mécanisme d'indexation ainsi que les méthodes permettant d'évaluer les réponses produites.

Cette limite rejoint finalement le problème auquel le projet cherchait à répondre : une intelligence artificielle ne crée pas la mémoire d'une organisation à sa place. Elle peut faciliter l'exploitation d'une procédure, d'un rapport ou d'une décision correctement capitalisés ; elle ne peut retrouver une connaissance qui n'a jamais été conservée.

Le projet laisse néanmoins deux résultats tangibles : une proposition argumentée et validée à plusieurs niveaux de la direction, ainsi qu'un cahier des charges permettant d'envisager sa reprise lorsque les conditions d'infrastructure seront réunies. À cela s'ajoute un apport plus personnel : l'expérience d'avoir transformé une idée issue de ma formation en une proposition professionnelle et de l'avoir défendue devant les acteurs appelés à en apprécier la valeur.

Avec ce dernier chantier se referme la présentation des travaux menés au cours de l'apprentissage. Leurs niveaux d'aboutissement diffèrent, mais ils permettent désormais de dresser un bilan plus transversal des contributions apportées à la collectivité, de leurs limites et des conditions nécessaires à leur pérennisation.

BILANS COLLECTIVITÉ ET ÉTUDIANT

Mon apprentissage a progressivement dépassé la seule réalisation de missions techniques. Il m'a conduit à intervenir sur plusieurs dimensions structurantes du système d'information : développement de solutions, conception et analyse d'architectures, cybersécurité, formalisation de processus, production d'indicateurs et analyse de solutions destinées à éclairer des décisions techniques ou budgétaires.

Cet environnement a ainsi constitué un terrain particulièrement favorable pour mobiliser et approfondir une quadruple compétence en développement logiciel, architecture, cybersécurité et gestion de projet. Les missions réalisées m'ont régulièrement amené à croiser ces quatre dimensions : concevoir une solution ne suffisait pas, il fallait également réfléchir à son intégration dans le SI, aux risques qu'elle introduisait, aux acteurs concernés, aux contraintes de mise en œuvre et aux conditions permettant d'en assurer le suivi dans le temps.

04

4.1. BILAN POUR LA COLLECTIVITÉ

Contributions et transformations apportées

Mon activité m'a conduit à intervenir sur la gouvernance du système d'information, la formalisation des pratiques, l'aide à la décision, la cybersécurité, la production d'indicateurs et l'automatisation de certains traitements. Chacun de ces domaines recouvre certaines des réalisations présentées dans ce mémoire.

Une part importante de mon travail a porté sur la mise sous contrôle de processus transversaux du SI. Plus de cinq processus ont ainsi été formalisés puis validés, notamment autour des habilitations, des arrivées, mobilités et départs des agents, ainsi que de l'autorisation des applications pouvant être installées. Ces travaux ont permis de transformer des pratiques parfois dispersées en règles de fonctionnement explicites, avec des responsabilités, des points de validation et des critères de décision clairement identifiés.

Tableau 4.1 — Principales formes de contribution à la collectivité

Domaine	Contributions principales	Valeur recherchée
Gouvernance et processus	Plus de cinq processus validés : habilitations, arrivées/mobilités/départs, validation d'applications et autres processus transversaux	Clarifier les responsabilités, homogénéiser les pratiques et améliorer la traçabilité
Aide à la décision	Analyses comparatives, études de solutions, sourcing, avis techniques sur l'opportunité de nouvelles acquisitions	Fournir des éléments techniques, fonctionnels et économiques avant engagement de ressources
Pilotage du SI	Construction ou proposition de nouveaux indicateurs sur les incidents, menaces, actifs et autres dimensions du SI	Transformer certaines données en éléments utilisables pour le pilotage et la décision
Développement et automatisation	Référentiel applicatif, CyberAlerte, calculateur automatisé de factures et autres outils	Réduire les traitements manuels, fiabiliser certaines opérations et répondre à des besoins internes
Capitalisation	Procédures, documentations, cahiers des charges, guide de sourcing, modèles et règles réutilisables	Permettre la reprise des travaux et réduire la dépendance aux connaissances individuelles
Services numériques	Contribution aux travaux relatifs à la plateforme open data de la Guadeloupe	Participer à la mise à disposition et à la valorisation des données publiques

Certaines réalisations ont apporté des gains directement opérationnels. Pour le référentiel applicatif, la fiabilité obtenue (94,2 % de classifications correctes) permet surtout de disposer d'une vision du patrimoine logiciel suffisamment robuste pour être utilisée dans des décisions concrètes : identifier les équipements concernés par une version, apprécier l'hétérogénéité du parc ou préparer une action de mise à jour, sans devoir retraiter manuellement des milliers d'entrées.

Avec CyberAlerte, l'apport se situe davantage dans la capacité de traitement de la cellule cybersécurité. La réduction du temps et du nombre de manipulations nécessaires (86 % de gain de temps) permet de consacrer moins de ressources aux opérations répétitives et de libérer du temps pour les tâches nécessitant réellement une analyse humaine et améliore la reproductibilité de la réponse à incident.

D'autres contributions créent une valeur moins immédiatement quantifiable. Une analyse peut conduire à recommander une acquisition, mais elle peut également permettre de l'écarter lorsqu'elle apparaît inadaptée au besoin, trop coûteuse ou insuffisamment maîtrisée. Dans les deux cas, la valeur produite permet d'éclairer une décision avant qu'elle ne génère des coûts ou des dépendances supplémentaires.

La même logique s'applique aux indicateurs développés ou envisagés. Plusieurs d'entre eux visent à améliorer la visibilité sur les incidents de sécurité, les menaces ou l'état du système d'information afin que certaines décisions puissent davantage s'appuyer sur des données observables. Des exemples de ces indicateurs sont présentés en annexe E.

Pris dans leur ensemble, ces travaux peuvent être résumés par quelques transformations :
formaliser ce qui était implicite ; structurer ce qui était dispersé ; mesurer ce qui était difficilement observable ; automatiser ce qui était répétitif ; documenter ce qui devait pouvoir être transmis ; et apporter des éléments de preuve là où une décision devait être prise.

Principes et pratiques apportés au cours des travaux

Au-delà des réalisations elles-mêmes, plusieurs principes ont progressivement guidé ma manière d'aborder les missions. Ils constituent également des pratiques que j'ai cherché à intégrer aux solutions et méthodes proposées à la collectivité.

- Une décision ou une action doit pouvoir être comprise et reprise par quelqu'un d'autre que son auteur initial.
- Une solution ne devrait pas dépendre uniquement de la mémoire de celui qui l'a conçue. Documentation, historique, journalisation et traçabilité doivent donc être considérés comme une partie de la réalisation.
- Une affirmation doit autant que possible être accompagnée d'une preuve.
- Dire qu'une solution est plus rapide, plus fiable ou plus adaptée ne suffit pas : benchmarks, KPI, scénarios reproductibles, tests ou démonstrations doivent permettre de l'étayer.
- Exigence → choix → preuve.
- Un choix technique ou organisationnel doit répondre à une exigence identifiable, puis pouvoir être évalué selon des critères cohérents avec cette exigence.
- Le niveau de validation doit être proportionnel au risque.
- Une fonctionnalité susceptible d'avoir un impact sur un compte, une donnée ou un système critique exige davantage de contrôles qu'une opération facilement réversible.
- Les responsabilités importantes doivent être explicites.
- BPMN, RACI, propriétaires de données ou séparation entre demande, validation et exécution permettent de réduire les zones d'ambiguïté.
- Automatiser ce qui est déterministe ne signifie pas supprimer la décision humaine.
- L'automatisation est pertinente pour réduire les opérations répétitives ; les décisions engageant une responsabilité ou présentant un impact important doivent conserver les contrôles adaptés.
- Une solution temporaire doit rester identifiable comme temporaire.
- Un compromis peut être rationnel, mais il doit pouvoir être retrouvé, réévalué et remplacé lorsque son contexte de justification disparaît.
- Une idée ne suffit pas pour convaincre.
- La présentation de l'assistant documentaire m'a particulièrement appris qu'une proposition devient plus crédible lorsqu'elle arrive accompagnée de cas d'usage, d'éléments chiffrés, d'une architecture, de bénéfices attendus, mais aussi de contraintes, de risques et de limites.
- Une méthode n'a de valeur que si elle aide réellement à résoudre le problème.
- BPMN, matrices multicritères, analyses de risques, TCO, benchmarks ou cartes mentales ne doivent pas être utilisés pour eux-mêmes, mais lorsqu'ils permettent de mieux comprendre, décider, vérifier ou transmettre.

Ces principes m'ont permis de ne plus considérer la réussite d'une mission comme la seule livraison d'un artefact fonctionnel, mais comme la capacité à produire une solution justifiée, testable, compréhensible et reprenable. Toutes les réalisations ont ainsi été accompagnées d'une documentation adaptée, selon l'un des principes qui m'ont le plus marqué : une solution terminée doit pouvoir être comprise, maintenue et reprise sans dépendre de la présence de celui qui l'a conçue.

Pérenniser les apports

La valeur produite au cours de l'apprentissage dépend également de ce que ces travaux deviendront.

Quatre conditions apparaissent particulièrement importantes : un responsable identifié, capable d'arbitrer l'évolution d'un outil ou d'un processus ; une actualisation organisée, afin qu'une cartographie, une procédure ou un référentiel ne devienne pas progressivement obsolète ; la transmission des connaissances, pour éviter qu'une capacité critique repose sur une seule personne ; et enfin des ressources réelles, notamment du temps, des habilitations, une infrastructure et, lorsque nécessaire, un budget.

Les suites des différents travaux sont ainsi identifiables : poursuivre l'industrialisation du référentiel applicatif, valider les fonctionnalités de CyberAlerte encore limitées par les habilitations disponibles, faire évoluer le suivi actuel des droits vers une gouvernance IAM plus complète, réutiliser les méthodes de sourcing lors de futures consultations ou encore reprendre l'assistant documentaire lorsque les conditions d'infrastructure le permettront.

Cette question de la pérennité est importante. Un outil, une procédure ou une décision peut parfaitement répondre au besoin au moment où il est créé, puis devenir progressivement une contrainte si sa documentation, sa maintenance, son propriétaire ou sa remise en question sont oubliés.

Cette observation constitue l'un des enseignements majeurs de l'apprentissage. Avant de l'approfondir sous l'angle de la dette organisationnelle, le chapitre suivant revient sur les apports de cette expérience à mon parcours, à mes compétences et à l'évolution de mon positionnement professionnel.

4.2. Bilan pour l'étudiant-apprenti

La formation MIAGE comme socle adaptable

La formation MIAGE a constitué un socle particulièrement adapté aux missions rencontrées parce qu'elle associe plusieurs disciplines qui, en entreprise, apparaissent rarement de manière isolée. Développement, données, architecture, systèmes d'information, gestion de projet, modélisation des processus ou gouvernance se sont ainsi retrouvés simultanément dans plusieurs travaux.

Les enseignements de développement et de gestion des données ont naturellement servi à concevoir le référentiel applicatif ou certaines automatisations. Les connaissances relatives aux systèmes d'information ont permis de replacer ces développements dans un environnement plus large, en tenant compte des acteurs, des flux, des applications et de leurs dépendances. La modélisation des processus a trouvé une application concrète dans le traitement des incidents ou la gouvernance des habilitations, tandis que les enseignements de gestion de projet ont accompagné le cadrage, l'organisation des travaux et l'identification des risques.

L'intérêt de la formation m'est toutefois apparu davantage dans la combinaison de ces connaissances que dans leur utilisation séparée.
Une situation réelle ne se présentait jamais sous la forme : « il faut appliquer un RACI » ou « il faut réaliser une analyse de risques ».

Elle se présentait plutôt comme un problème : des responsabilités peu claires, une décision difficile à justifier, une information dispersée ou une opération trop dépendante d'une personne. Il fallait alors reconnaître qu'un outil ou une méthode appris précédemment pouvait aider, puis l'adapter aux contraintes rencontrées.

Cette expérience m'a donc permis de comprendre que les enseignements de la MIAGE fournissent avant tout des fondamentaux mobilisables selon le contexte. Cette logique explique pourquoi certaines méthodes utilisées dans les missions ont progressivement évolué. Une modélisation BPMN n'avait pas besoin du même niveau de détail selon qu'elle servait à comprendre un processus ou à préparer son automatisation. De même, une matrice de décision, un benchmark ou une analyse de risques n'avaient d'intérêt que s'ils permettaient effectivement d'éclairer une décision.

J'ai ainsi progressivement appris à utiliser une méthode pour ce qu'elle apporte et non simplement parce que je la connaissais.

Compétences consolidées et évolution vers une posture d'ingénierie

Les missions ont naturellement renforcé plusieurs compétences techniques : développement, PowerShell, automatisation, exploitation de données, architecture applicative, cybersécurité, IAM, tests ou encore conception d'une architecture documentaire fondée sur l'intelligence artificielle. L'apport le plus important dépasse cependant l'apprentissage de technologies supplémentaires.
J'ai progressivement appris à aborder une solution comme un ensemble de questions :

- Quel problème cherche-t-on réellement à résoudre ?
- Quelles contraintes doivent être respectées ?
- Quels sont les risques ?
- Quelles solutions sont possibles ?
- Pourquoi choisir celle-ci plutôt qu'une autre ?
- Comment vérifier qu'elle fonctionne ?
- Comment démontrer qu'elle apporte réellement quelque chose ?
- Qui pourra la maintenir après sa livraison ?

Cette manière de raisonner marque pour moi le passage d'une logique principalement centrée sur la réalisation à une posture davantage orientée ingénierie.
Développer le référentiel ne consistait ainsi plus seulement à écrire une application, mais à réfléchir à l'architecture, au modèle de données, à la fiabilité de la normalisation, à ses performances, aux usages possibles et à sa maintenance. Automatiser le traitement d'un incident ne consistait pas uniquement à écrire un script PowerShell, mais à formaliser le processus, identifier les opérations sensibles, définir les comportements en cas d'erreur et construire un environnement permettant de tester réellement les effets produits.

Cette évolution m'a également appris à accepter qu'une solution techniquement intéressante ne soit pas nécessairement la bonne solution à un instant donné. CKAN ou l'assistant documentaire l'ont montré différemment : les contraintes d'organisation, d'infrastructure ou de disponibilité peuvent rendre pertinente une interruption, un report ou une solution transitoire.

La capacité d'ingénierie réside donc aussi dans la capacité à arbitrer, et non uniquement à réaliser.

Initiative, autonomie et communication comme leviers de valeur

L'initiative constitue l'un des aspects les plus importants de mon apprentissage. Mon maître d'apprentissage m'a accordé rapidement une autonomie importante, ce qui m'a permis de ne pas limiter les missions à leur formulation initiale et de proposer régulièrement des prolongements lorsque j'identifiais une possibilité d'amélioration.

La demande portant sur la disparité des versions logicielles aurait par exemple pu être satisfaite par la production du fichier Excel demandé. J'ai proposé d'en faire un référentiel davantage pérenne. La procédure de signalement d'incident a conduit à une version plus visuelle puis à une réflexion sur son automatisation. Le sourcing IAM a donné lieu à la création d'un modèle destiné à être réutilisable. Enfin, le projet d'assistant documentaire est directement né d'une initiative consistant à transposer un travail académique à une problématique que j'avais identifiée dans la collectivité.

Ces expériences m'ont appris que l'autonomie implique d'être capable d'identifier un problème, de proposer une orientation, d'en mesurer les conséquences et de savoir quand revenir vers les personnes disposant de la responsabilité décisionnelle, et l'initiative doit donc rester accompagnée d'une capacité à convaincre.

La présentation de l'assistant documentaire devant l'ensemble de la DTN, en présence notamment du DSI et de membres de l'encadrement, a été particulièrement formatrice à cet égard. Il ne suffisait pas d'affirmer qu'une technologie était prometteuse. Il fallait expliquer quel problème elle pouvait résoudre et pourquoi elle méritait que l'organisation y consacre du temps. J'avais donc préparé des cas d'usage concrets, des données statistiques, une architecture envisagée ainsi que les bénéfices, contraintes et risques identifiés.

Cette expérience m'a appris une règle professionnelle que je considère désormais essentielle :
Une idée devient beaucoup plus difficile à écarter lorsqu'elle arrive accompagnée d'un problème démontré, d'une proposition structurée et d'éléments permettant de la discuter.
Le travail avec des responsables, des utilisateurs et des prestataires a également renforcé ma capacité à adapter mon niveau de discours. Une architecture ne se présente pas de la même manière à un technicien, à un responsable métier ou à un décideur. La pertinence d'une solution ne suffit pas : il faut également être capable de la rendre compréhensible aux personnes dont dépend sa réalisation.
Cette combinaison entre initiative, autonomie, responsabilité et communication correspond à l'évolution professionnelle la plus importante que je retiens de cette année.

Apports personnels et contribution au projet professionnel

L'apprentissage a modifié ma manière de raisonner au-delà des seules missions professionnelles. L'étude des systèmes d'information m'a notamment appris à accorder davantage d'importance à la notion de **système** elle-même. Un résultat dépend rarement d'une action isolée : il résulte d'un ensemble de ressources, de règles, d'acteurs, de contraintes et de mécanismes de retour qui interagissent. Cette manière de raisonner m'a amené à retrouver des logiques similaires dans des situations beaucoup plus générales, y compris personnelles. L'analyse de risques peut par exemple structurer une décision en distinguant probabilité, impact et moyens de réduction ; la gestion de projet donne des méthodes pour atteindre ses objectifs ; une carte mentale aide à décomposer un problème avant de décider. Tout cela peut être transposé à notre propre quotidien.

Cette transversalité constitue l'un des aspects de la MIAGE que je comprends mieux à l'issue de mon parcours. Plus que des technologies ou une succession de méthodes, la formation m'a fourni des outils pour structurer un problème et raisonner sur des systèmes complexes.

Avec le temps, j'ai cependant constaté qu'une difficulté apparaissait : connaître de nombreuses méthodes ne signifie pas nécessairement savoir laquelle mobiliser face à un problème donné. J'ai donc commencé à les organiser dans un référentiel méthodologique personnel, que j'enrichis progressivement à partir des enseignements reçus, des retours d'expérience de mes professeurs et de mes propres missions. Celui-ci regroupe les méthodes par grands domaines, par exemple gestion de projet, ingénierie logicielle, développement, évaluation, puis par types de besoins plus précis.
Pour éviter que ce référentiel ne devienne une simple liste, je l'ai associé à plusieurs arbres de décision. Lorsqu'un problème se présente, ceux-ci permettent de partir de la nature du besoin, puis d'affiner progressivement selon le contexte et les contraintes jusqu'à faire émerger une ou plusieurs méthodes pertinentes. Le choix final reste ensuite un arbitrage : l'arbre aide à réduire les possibilités et à justifier plus clairement la méthode retenue.

Une dernière couche permet de conserver succinctement le contexte du choix, les raisons de l'arbitrage et le retour d'expérience obtenu après utilisation. L'objectif est de pouvoir répondre, a posteriori, à une question simple : était-ce réellement la bonne méthode dans cette situation ? Une difficulté rencontrée peut alors conduire à enrichir le référentiel, modifier une branche de l'arbre ou, au contraire, confirmer la pertinence d'un choix pour un contexte similaire.

Cette démarche représente pour moi un lien concret entre **la théorie et la pratique** : apprendre des méthodes, savoir les sélectionner, les confronter au terrain puis capitaliser ce que leur mise en œuvre m'a réellement appris. Elle continue d'évoluer à mesure de mes expériences et constitue probablement l'un des acquis que je souhaite le plus conserver après la formation.

Cette année a également changé la manière dont j'appréhende mes propres compétences. Il est relativement simple d'affirmer sur un CV savoir développer une application, analyser un besoin ou concevoir une architecture, mais à l'issue de cet apprentissage, je souhaite davantage pouvoir répondre à ces affirmations par une autre question : **quelles preuves suis-je capable d'apporter ?**

Les réalisations présentées dans ce mémoire constituent une partie de cette réponse. Parmi elles, une compétence me paraît particulièrement représentative de mon évolution : la capacité à concevoir une solution dans son ensemble. Il ne s'agit plus seulement de développer une fonctionnalité, mais de **partir d'un besoin, le traduire en exigences, envisager plusieurs réponses, mobiliser les méthodes adaptées pour les comparer puis justifier les compromis retenus**.

Cette réflexion impose également de regarder au-delà de la **livraison immédiate**. Une solution destinée à durer doit être pensée en considérant des propriétés telles que la sécurité, la maintenabilité, le coût, ou la capacité d'évolution, sans pour autant construire dès le départ une architecture surdimensionnée. L'enjeu consiste davantage à identifier quels compromis peuvent être acceptés aujourd'hui et lesquels deviendraient particulièrement coûteux à remettre en cause demain. C'est une leçon que j'ai particulièrement appliquée au cours de mes années d'alternance.

Cette manière de raisonner explique en partie mon intérêt croissant pour les métiers de l'architecture, et plus particulièrement ceux d'architecte logiciel ou d'architecte cybersécurité. Ils nécessitent précisément de dépasser la seule maîtrise technique pour traduire les besoins, analyser les risques et les contraintes, comparer plusieurs possibilités et argumenter les décisions prises.

Plus largement, cette année m'a fait évoluer d'une posture où la réussite d'une mission pouvait être résumée par « j'ai réalisé ce qui m'était demandé » vers une interrogation différente :
« **Ai-je réellement résolu le bon problème, puis-je démontrer pourquoi mes choix sont pertinents et la solution pourra-t-elle continuer à produire de la valeur après mon intervention ?** »

Certaines difficultés rencontrées trouvaient leurs origines dans des solutions temporaires peu à peu installées dans la durée. Ce décalage entre le bénéfice immédiat d'un choix et ses conséquences futures constitue précisément le point de départ du dernier chapitre, consacré à l'accumulation de ces choix sous l'angle de la dette organisationnelle.

SUJET DE RÉFLEXION

LA DETTE ORGANISATIONNELLE : QUAND LES RACCOURCIS D'AUJOURD'HUI DEVIENNENT LES FREINS DE DEMAIN

Les difficultés rencontrées au cours des missions précédentes ne relevaient pas toutes de la même nature. Certaines étaient techniques, d'autres liées à la documentation, aux données, aux processus, aux responsabilités ou encore à la concentration de certaines connaissances. Pourtant, plusieurs présentaient une caractéristique commune : leur coût n'était pas nécessairement visible au moment où elles apparaissaient.

Une configuration insuffisamment documentée peut fonctionner pendant plusieurs années. Un traitement manuel peut rester acceptable tant que son volume est faible. Une habilitation peut rester attribuée sans provoquer immédiatement d'incident. Une connaissance détenue par une seule personne ne devient problématique que lorsque cette personne n'est plus disponible.

Ces situations conduisent à s'interroger sur la manière dont des choix ou des pratiques initialement acceptables peuvent progressivement devenir des contraintes pour l'organisation. La problématique de ce mémoire est précisément :

> **Comment l'accumulation de choix à court terme transforme-t-elle le temps gagné aujourd'hui en dette organisationnelle freinant les systèmes d'information de demain ?**

La littérature sur la dette technique fournit un premier cadre pour analyser ce déplacement du coût dans le temps. D'autres travaux ont depuis étendu cette logique aux dimensions architecturales, documentaires, sociales ou organisationnelles. Ces différentes approches soulèvent cependant plusieurs questions : ces dettes peuvent-elles être analysées indépendamment ? Comment finissent-elles par s'enraciner dans une organisation ? Sous quelles formes leurs coûts réapparaissent-ils ? Et surtout, comment éviter qu'un compromis temporaire ne devienne progressivement une contrainte structurelle ?

Ce chapitre cherchera à répondre à ces questions en confrontant la littérature aux situations observées au cours de l'apprentissage. L'objectif sera finalement de proposer une lecture opérationnelle de la dette organisationnelle appliquée aux systèmes d'information, ainsi qu'un cadre permettant d'en identifier, prévenir et piloter les principales manifestations.

05

De la dette technique à l'élargissement de la métaphore

La notion de dette appliquée aux systèmes d'information trouve son origine dans la dette technique. La métaphore introduite par Ward Cunningham au début des années 1990 permet d'expliquer qu'un compromis facilitant le développement à court terme peut augmenter ultérieurement l'effort nécessaire pour maintenir ou faire évoluer un logiciel.

Cette logique présente un intérêt particulier pour la problématique étudiée ici : elle introduit l'idée qu'un **gain immédiat peut déplacer une partie de son coût dans le futur**.

Les travaux consacrés à la dette technique montrent que ce phénomène ne reste pas marginal. Une famille d'enquêtes menée auprès de 653 professionnels dans six pays identifie la pression temporelle et les délais comme sa cause la plus fréquemment citée, tandis que le rework, les retards et la dégradation de la maintenabilité figurent parmi ses conséquences les plus signalées. Cette étude estime également qu'environ **25 % de l'effort de développement** peuvent être consacrés aux difficultés induites par la dette technique.

Mais les difficultés rencontrées dans un système d'information ne proviennent pas uniquement du code.

La littérature a progressivement étendu la métaphore. Des travaux portent ainsi sur la dette architecturale, lorsque des choix de conception limitent les évolutions futures ; sur la dette documentaire, lorsque des informations manquantes ou obsolètes compliquent la compréhension et l'intégration ; ou encore sur la dette sociale, qui s'intéresse aux coûts issus de certaines structures de collaboration et relations entre acteurs.

Cette diversification montre que le mécanisme de report d'un coût futur peut concerner **plusieurs composantes du fonctionnement d'une organisation.**

Plus récemment, la notion de dette organisationnelle cherche précisément à élargir cette réflexion. La revue d'Al-Baik et al. met notamment en avant des structures, politiques ou processus devenus inadaptés, ainsi que des phénomènes de silos, de coordination ou de partage des connaissances. Les conséquences identifiées concernent entre autres la productivité, l'agilité et l'innovation.

Le concept reste cependant émergent et ses frontières ne sont pas encore totalement stabilisées. Les différentes formes de dette sont souvent étudiées séparément, alors que le fonctionnement d'un système d'information repose justement sur leurs interactions : les technologies dépendent des processus, les processus des responsabilités, les responsabilités de la connaissance disponible et l'ensemble de la qualité de l'information utilisée.

Des frictions organisationnelles aux coûts mesurables

Une difficulté de ce concept de dette tient au fait que ses conséquences sont **rarement présentées comme de vraies pertes importantes** dans les comptes d'une organisation. Ce coût apparaît sous d'autres formes : temps passé à rechercher une information, travail refait, maintenance, retard, attente d'un interlocuteur, projet reporté ou incident plus difficile à maîtriser, souvent perçus comme des éléments peu révélateurs.

Pourtant plusieurs études mentionnées dans le tableau 5.1 permettent néanmoins d'en apprécier les ordres de grandeur.

Tableau 5.1 – Ordres de grandeur associés à différentes formes de friction organisationnelle

Phénomène	Résultat documenté	Lecture pour la dette organisationnelle
Recherche d'information	McKinsey estimait que les travailleurs fortement dépendants de l'information consacraient environ 19 % de leur temps à rechercher et rassembler celle-ci.	Une information existante mais difficilement accessible mobilise une capacité qui ne produit pas directement de nouvelle valeur.
Partage insuffisant des connaissances	Une enquête Panopto/YouGov auprès de 1 001 travailleurs américains estimait à 5,3 h par semaine le temps consacré à attendre une information ou reconstruire une connaissance existante, et modélisait jusqu'à 47 M$ de productivité annuelle perdue pour une grande entreprise américaine moyenne de son modèle.	La connaissance détenue uniquement par certaines personnes constitue un coût lorsque ces personnes ne sont pas disponibles.
Frictions de développement	En 2025, Atlassian rapportait que 50 % des développeurs interrogés perdaient plus de 10 h par semaine dans différentes inefficacités ; ses travaux placent notamment la recherche d'information, la documentation insuffisante et la dette technique parmi les sources de friction.	Les effets de plusieurs dettes peuvent s'additionner dans la même journée de travail.
Dette technique	Une enquête internationale auprès de 653 professionnels retrouve le rework, les retards et la maintenabilité parmi les effets les plus fréquents de la dette technique.	L'économie réalisée lors de la création du système peut être remboursée lors de chacune de ses évolutions.
Mauvaise performance projet	Le Pulse of the Profession 2021 de PMI évaluait à 9,4 % la part des investissements gaspillée du fait d'une mauvaise performance des projets dans l'échantillon étudié.	Les défauts de processus, de coordination ou de décision possèdent également une traduction économique.
Incident de cybersécurité	IBM évalue en 2026 le coût moyen mondial d'une violation de données à 4,99 M$, sur la base de 602 organisations touchées étudiées avec le Ponemon Institute.	Une dette ne provoque pas nécessairement l'incident, mais des actifs, droits ou procédures mal maîtrisés peuvent en compliquer la prévention ou la résolution.

Ces chiffres ne mesurent pas un coût pour la Région Guadeloupe, et leurs périmètres, populations et méthodologies diffèrent. Ils montrent en revanche que les phénomènes auxquels cette dette peut contribuer : recherche d'information, perte de connaissance, rework, mauvaises performances de projet ou exposition cyber, peuvent atteindre des ordres de grandeur économiques significatifs.

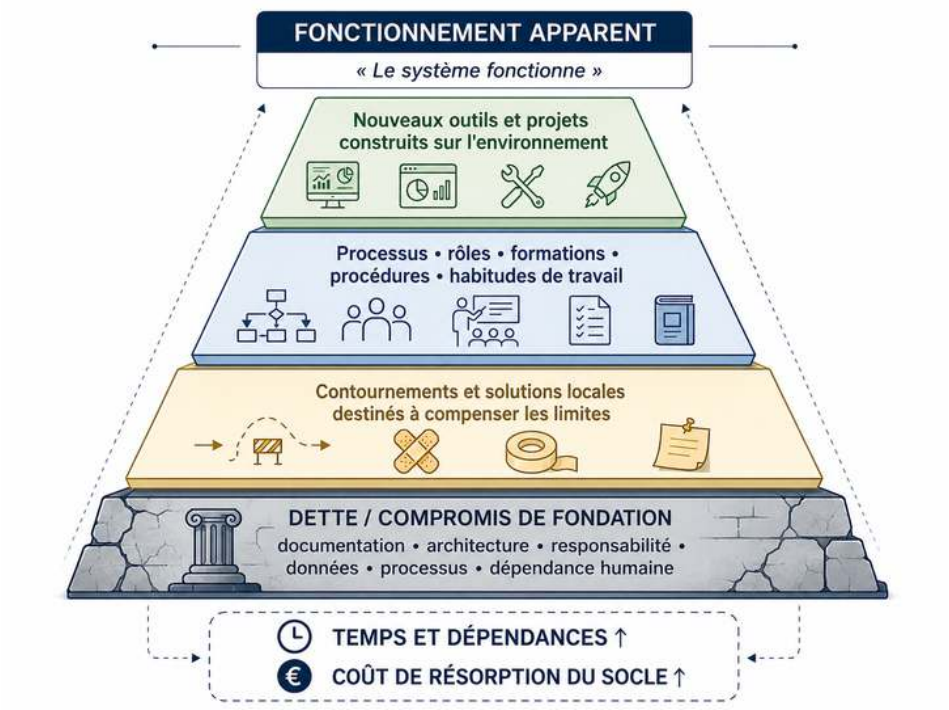
Ils permettent surtout de dépasser l'idée qu'une documentation insuffisante ou une opération manuelle ne représente que « quelques minutes perdues ». Au contraire, cela montre que quelques minutes peuvent devenir un coût lorsqu'elles doivent être p**ayées par plusieurs personnes, plusieurs fois, pendant plusieurs années**.

Du raccourci à l'enracinement : la pyramide de dette

J'ai constaté en revanche que ces formes de dettes ne sont pas difficiles à supprimer uniquement parce qu'elles vieillissent. Elles deviennent difficiles à supprimer aussi parce que **d'autres éléments commencent à dépendre d'elles.**

Je propose de représenter ce phénomène sous la forme d'une pyramide.

Figure 5.1 — Pyramide d'enracinement de la dette organisationnelle



Une modification insuffisamment documentée peut initialement être simple à corriger. Mais si elle reste en place, une procédure peut être construite autour d'elle. Les agents apprennent cette procédure. Un fichier est créé pour suivre ses exceptions. Un autre outil utilise ensuite ce fichier. Une automatisation vient enfin faciliter ce fonctionnement.

Le problème initial se retrouve progressivement sous plusieurs couches de dépendances.

Cette dynamique rejoint la théorie de la dépendance au sentier. Sydow, Schreyögg et Koch décrivent comment des mécanismes auto-renforçants peuvent progressivement réduire l'espace de choix d'une organisation jusqu'à produire une situation de lock-in. Les travaux sur les routines organisationnelles montrent également qu'avec le temps, leur interdépendance et leur insertion dans les structures de l'organisation peuvent renforcer leur persistance.

Cela explique un paradoxe important : une dette fortement enracinée **peut ne plus être perçue comme un problème.**

Le système fonctionne, les agents savent comment l'utiliser, des procédures existent, des outils ont même été créés pour faciliter son fonctionnement.

La question n'est alors plus : « Pourquoi fonctionnons-nous ainsi ? » mais : « **Comment améliorer encore cette manière de fonctionner ?** »

Il devient possible de tenter de “perfectionner” **les conséquences d’un problème** au lieu d’interroger sa cause.
On peut ainsi observer une succession du type :
problème initial
→ contournement A
→ limitation de A
→ outil B pour suivre A
→ procédure C pour gérer B
→ contrôle D parce que C n’est pas fiable
→ automatisation E pour accélérer D

alors que la question architecturale pourrait être : Faut-il encore conserver A ?

Chaque décision peut être rationnelle localement, puisqu’elle répond au système tel qu’il existe à cet instant. Mais leur accumulation peut produire globalement une organisation sous-optimale.

C’est l’un des mécanismes qui rendent la dette organisationnelle particulièrement difficile à résorber : corriger la fondation impose progressivement de modifier ce qui a été construit au-dessus.

Le temps confisqué : l’intérêt payé par l’organisation

La métaphore de la dette me paraît particulièrement pertinente parce que le temps intervient des deux côtés du mécanisme.
Il est d’abord ce que l’organisation cherche souvent à économiser « aujourd’hui » :
- ne pas documenter maintenant ;
- ne pas automatiser maintenant ;
- ne pas normaliser maintenant ;
- ne pas revoir les droits maintenant ;
- utiliser une solution transitoire plutôt que lancer immédiatement un projet plus complexe.

Mais il est également ce que l’organisation **finit fréquemment par payer.**

Tableau 5.2 – Relation entre les formes de dette et le temps consommé

Dette	Forme de temps ultérieurement consommé
Documentaire	recherche, compréhension, reconstruction
Informationnelle	nettoyage, rapprochement, vérification
Processuelle	manipulation répétée, rework
Humaine	attente, transfert, montée en compétence
Gouvernance	coordination, validation, arbitrage
Technique	maintenance, contournement, refactoring

Je propose de désigner par « temps confisqué » cette part de capacité qui n'est plus mobilisée pour créer une nouvelle valeur, mais pour supporter les conséquences d'obligations antérieures.

Cette lecture permet de rapprocher la dette organisationnelle de la métaphore financière :
- gain initial : temps ou ressources économisés lors du compromis ;
- capital à rembourser : travail qu'il faudra réaliser un jour pour corriger la cause de la dette ;
- intérêts temporels : travail supplémentaire à effectuer tant qu'elle n'est pas corrigée ;
- risque : événements défavorables dont la dette peut augmenter la probabilité ou l'impact ;
- coût d'opportunité : projets ou améliorations auxquels l'organisation renonce faute de capacité disponible.

Le temps possède ainsi une place particulière : il est à la fois ce que le compromis cherche à économiser et une partie de ce que l'organisation finit par rembourser.

Lorsque cette charge peut être mesurée, elle possède également une traduction économique :
$Coût_{temps} \approx Temps_{confisqué} \times Coût_{horaire\ chargé}$

Cette estimation ne représente toutefois qu'une partie du coût réel, puisqu'elle ne prend pas directement en compte les incidents, les projets retardés, les opportunités abandonnées ou certains impacts financiers et réputationnels.
Les situations rencontrées pendant l'apprentissage permettent d'observer différentes formes de ce mécanisme.

Tableau 5.3 — Manifestations et conséquences des dettes observées

Situation	Dette dominante	Intérêt ou contrainte observable	Conséquence potentielle sur l'évolution du SI
CKAN	Documentaire + humaine + gouvernance	Diagnostic supplémentaire puis dépendance bloquante	Reprise et évolution de la solution compromises
Patrimoine logiciel	Informationnelle + technique	Recherche, rapprochement et traitement de l'hétérogénéité	Patching, migration et analyse de vulnérabilités complexifiés
CyberAlerte	Processuelle	Répétition d'opérations manuelles	Capacité de la cellule absorbée par des tâches répétitives
IAM	Gouvernance + informationnelle	Recherche, justification et révision difficiles des droits	Privilèges persistants et automatisation du cycle de vie limitée

La conséquence la plus importante apparaît cependant lorsque ces dettes rencontrent les transformations futures du SI. Les technologies nouvelles ne repartent pas d'un environnement vierge : elles héritent de la qualité des données, processus, droits, architectures et connaissances existantes.

Et surtout, ce mécanisme peut **s'autoalimenter** : la dette consomme le temps qui aurait permis de la résorber. Une équipe absorbée par le maintien, la recherche et les contournements dispose de moins de capacité pour documenter, automatiser ou refactorer. Cela favorise de nouveaux compromis à court terme. La littérature sur la dette organisationnelle retrouve précisément des effets sur la productivité et la capacité d'adaptation, tandis que les travaux sur la dette technique identifient la réduction de productivité et l'augmentation des coûts de maintenance parmi ses conséquences importantes.

Deux effets supplémentaires apparaissent donc : la dette mobilise des ressources pour maintenir l'existant et réduit celles qui restent disponibles pour préparer l'avenir.

Prévenir, détecter, résorber et piloter la dette

L'objectif ne peut toutefois pas être de supprimer toute dette. Les travaux récents sur la dette technique rappellent qu'elle peut résulter d'un choix délibéré d'expérience et qu'un système réaliste de gestion doit permettre de l'identifier et de la prioriser plutôt que prétendre l'éviter systématiquement.
La question devient donc : **Comment empêcher une dette utile à court terme de devenir invisible puis structurelle ?**

Tableau 5.4 — Registre de bonnes pratiques contre la dette organisationnelle

Dimension	Prévenir	Détecter	Résorber
Documentaire	Owner, source de référence, documentation intégrée aux livrables	Documents anciens, absents ou sans propriétaire ; temps de recherche	Mise à jour, consolidation, génération/tests documentaires
Informationnelle	Modèle, règles de qualité, propriétaire de données	doublons, données inconnues, incohérences, temps de rapprochement	nettoyage, normalisation, référentiel
Processuelle	BPMN, responsabilités, critères d'entrée/sortie	temps de cycle, exceptions, rework, opérations manuelles	simplification, standardisation, automatisation
Humaine	binômage, suppléance, transmission	compétence détenue par une seule personne, délais d'attente	formation croisée, documentation, transfert
Gouvernance	RACI, ownership, séparation des rôles, critères de décision	droits non revus, décisions sans propriétaire, validations anciennes	revue, réattribution, retrait ou nouveau processus
Technique	architecture, ADR, standards, tests, observabilité	backlog, incidents, obsolescence, coûts de maintenance	refactoring, simplification, remplacement, décommissionnement

Cette approche rejoint plusieurs référentiels existants. ISO 30401 traite la gestion de la connaissance comme un système à établir, maintenir, revoir et améliorer. Le NIST CSF 2.0 place quant à lui les rôles, responsabilités, actifs, politiques et mécanismes de gouvernance au cœur de la gestion du risque cyber. Dans le domaine logiciel, une revue systématique identifie notamment la visibilité de la dette et le refactoring parmi les stratégies les plus significatives de gestion de dette technique.

À partir des mécanismes identifiés précédemment, je propose un cadre simple permettant de faire émerger puis traiter une dette organisationnelle potentielle :
Détecter → Préqualifier → Analyser → Prioriser et traiter → Réévaluer

1. Détecter

La démarche part des frictions du quotidien, et non d'une dette supposée. Une banque de questions peut être utilisée individuellement ou collectivement :

- Est-ce que je perds régulièrement du temps à retrouver une information ?
- Une activité dépend-elle d'une seule personne ?
- Une solution temporaire est-elle devenue permanente ?
- Quelles tâches dois-je régulièrement refaire ou contourner ?
- Qu'est-ce qui me fait perdre du temps de manière répétée ?

Lorsque la réponse appelle une recherche de cause, les 5 pourquoi[26] peuvent être mobilisés immédiatement.

2. Préqualifier

Toute difficulté n'est pas une dette. Un arbre de décision permet de vérifier rapidement :

1. Le problème persiste-t-il ou se reproduit-il ?
2. Provient-il vraisemblablement d'un élément ancien insuffisamment traité, maintenu ou réévalué ?
3. Génère-t-il aujourd'hui du temps supplémentaire, un risque, du rework ou un blocage ?
4. Sa correction nécessite-t-elle désormais un effort spécifique ?

Si ces conditions sont réunies, la situation peut être considérée comme une dette potentielle.

3. Analyser et enregistrer

Pour un problème simple, les 5 pourquoi peuvent suffire. Pour une situation multicausale, un diagramme d'Ishikawa[27] adapté au SI peut explorer six dimensions :
processus · information/documentation · outils/architecture · humain/compétences · gouvernance · historique/contraintes
L'analyse alimente ensuite un registre comprenant :
signal initial · domaine concerné · type de dette · causes · origine/compromis · preuves · dépendances · statut
Le registre sert ainsi moins à découvrir la dette qu'à éviter qu'une dette identifiée ne redevienne invisible.

4. Prioriser et traiter

La mesure reste facultative et proportionnée au besoin. Lorsque cela aide à décider, peuvent être appréciés : effort de résorption, intérêts temporels, risque, âge, propagation, concentration humaine, coût d'opportunité et actionnabilité.

La priorité peut ensuite être déterminée en croisant : coût de conservation × effort de résorption

Tableau 5.5 — Matrice de priorisation de la dette

	Effort faible	**Effort élevé**
Coût de conservation élevé	Résorber rapidement	Planifier / sponsoriser
Coût de conservation faible	Corriger opportunément	Accepter / surveiller

Le traitement dépend de la cause identifiée : le tableau 5.4 précédent en fait référence.

5. Réévaluer

Enfin, la correction doit être vérifiée : la friction a-t-elle réellement diminué ? La cause a-t-elle été traitée ? Un nouveau contournement a-t-il été créé ?
Pour les situations simples, le cadre peut être réduit à :
Signal → Préqualification → Cause → Action → Responsable → Vérification
Le cadre complet n'est donc mobilisé que lorsque la complexité, le risque ou l'enracinement de la dette le justifient.

26. Méthode d'analyse consistant à demander successivement « pourquoi ? » afin de remonter progressivement de la conséquence observée vers sa cause racine

27. Outil d'analyse visuelle permettant d'identifier et de classer les causes possibles d'un problème afin d'en rechercher les causes profondes.

Portée, limites et réponse à la problématique

Cette réflexion a conduit à la création de plusieurs artefacts méthodologiques réutilisables : une banque de questions de détection, un arbre de préqualification, un modèle d'Ishikawa adapté à la dette organisationnelle, un registre de suivi, une matrice de priorisation ainsi qu'une matrice reliant les problèmes identifiés aux actions et livrables pouvant être mobilisés. Certains de ces artefacts sont présentés en annexe H.

Cependant, le cadre proposé reste exploratoire. Il repose sur un terrain unique et sur des situations auxquelles j'ai directement participé. Cette position de praticien donne accès aux mécanismes concrets, mais peut également influencer leur interprétation. Le recul temporel est en outre limité, alors qu'une dette peut ne produire ses principaux effets qu'après plusieurs années.

Les chiffres externes mobilisés dans ce chapitre ne constituent pas non plus une estimation des coûts de la collectivité. Ils servent à montrer que les phénomènes rapprochés de la dette organisationnelle peuvent avoir des conséquences économiques importantes dans d'autres contextes documentés.

Enfin, la typologie retenue, la pyramide d'enracinement, la notion de temps confisqué et le registre proposé constituent une interprétation personnelle issue du croisement entre le terrain et une littérature encore émergente. Al-Baik et al. soulignent eux-mêmes le besoin de recherches empiriques et quantitatives supplémentaires sur la dette organisationnelle.

Dans cette perspective, je propose de considérer la dette organisationnelle comme :

l'effet systémique produit par l'accumulation, l'interaction et l'enracinement de dettes techniques, documentaires, informationnelles, processuelles, humaines ou de gouvernance, issues de décisions, compromis ou obligations insuffisamment traités, et qui augmentent progressivement le temps, les ressources, les risques ou les contraintes nécessaires au fonctionnement et à l'évolution du système d'information.

Cette lecture apporte également une réponse à la problématique initiale :

> **L'accumulation de choix à court terme transforme le temps gagné aujourd'hui en dette organisationnelle lorsque les obligations différées persistent, commencent à produire des intérêts et donnent naissance à de nouvelles dépendances. Des dettes initialement locales, documentaires, techniques, humaines, informationnelles, processuelles ou de gouvernance, peuvent alors s'alimenter mutuellement et s'enraciner dans le fonctionnement de l'organisation. Leur résorption devient plus complexe à mesure que de nouveaux outils, processus et habitudes sont construits autour d'elles, tandis que l'organisation supporte entre-temps du temps de recherche, de rework, de maintenance, d'attente ou de coordination, ainsi qu'une exposition au risque et des opportunités retardées.**

La dette organisationnelle peut donc être comprise comme **l'effet systémique de plusieurs dettes qui s'accumulent et se renforcent.**

Cette lecture explique également pourquoi le terme de dette paraît particulièrement approprié. Comme dans une dette financière, un bénéfice est obtenu immédiatement, un capital reste à rembourser et le temps augmente le coût du maintien de l'obligation. Dans l'organisation, le temps possède même une place particulière : il est souvent à la fois ce que le compromis cherche initialement à économiser et ce que l'organisation finit par rembourser.

Or le temps humain possède lui-même un coût financier, tandis que les délais possèdent un coût d'opportunité. La solution pertinente n'est donc pas de supprimer tout raccourci. Une organisation doit pouvoir arbitrer, expérimenter et accepter temporairement certaines imperfections.

Mais avant de gagner une heure aujourd'hui, elle devrait être capable de se demander :

Combien d'heures cette décision risque-t-elle de reprendre demain ?

CONCLUSION

Mon apprentissage au sein de la Région Guadeloupe m'a permis d'intervenir sur des problématiques variées du système d'information, du développement à la cybersécurité, en passant par l'architecture, les processus et la gouvernance. Cette diversité a constitué une véritable opportunité : au-delà des solutions produites, elle m'a appris à formaliser avant d'automatiser, objectiver les décisions, documenter pour permettre la reprise et considérer les dépendances qui conditionnent la pérennité d'une solution.

La formation MIAGE a joué un rôle essentiel dans cette prise de recul. Son articulation entre informatique et gestion m'a permis d'aborder chaque mission non comme un problème uniquement technique, mais comme l'interaction entre technologies, données, processus, acteurs et décisions. Cette approche conforte aujourd'hui mon intérêt pour les métiers de l'architecture, où la maîtrise technique doit s'accompagner d'une capacité à arbitrer, anticiper et piloter la transformation, notamment grâce aux compétences issues de la gestion de projet.

Ce mémoire reste toutefois fondé sur un terrain unique et une période d'observation limitée. Certaines réalisations n'ont pu atteindre leur terme et les effets d'une dette organisationnelle peuvent nécessiter plusieurs années pour être pleinement observés. Ces limites ouvrent autant de perspectives pour approfondir les mécanismes identifiés et éprouver le cadre proposé dans d'autres organisations.

Enfin, penser au système d'information de demain conduit inévitablement à penser à l'intelligence artificielle, tant son intégration est devenue un enjeu majeur pour les organisations. Mais une IA, aussi performante soit-elle, reste dépendante des données, des connaissances, des processus, des responsabilités et des architectures auxquels elle accède.

À l'heure de cette course à l'IA, une dernière question demeure donc :

> **Et si le principal frein à l'intelligence artificielle de demain n'était pas sa technologie, mais la dette organisationnelle des systèmes d'information dans lesquels nous cherchons aujourd'hui à l'intégrer ?**

06

BIBLIOGRAPHIE

- AL-BAIK O. et al. (2024), « Organizational debt —Roadblock to agility in software engineering », PLOS ONE, 19(11).
- CUNNINGHAM W. (1992), « The WyCash Portfolio Management System », OOPSLA '92, ACM.
- HEVNER A. R. et al. (2004), « Design Science in Information Systems Research », MIS Quarterly, 28(1), p. 75-105.
- PEFFERS K. et al. (2007), « A Design Science Research Methodology for Information Systems Research », Journal of Management Information Systems, 24(3), p. 45-77.
- SYDOW J., SCHREYÖGG G. & KOCH J. (2009), « Organizational Path Dependence: Opening the Black Box », Academy of Management Review, 34(4), p. 689-709.
- ISO (2018), ISO 30401 — Knowledge management systems — Requirements.
- NIST (2024), The NIST Cybersecurity Framework (CSF) 2.0.
- GAO (2018), Data Protection: Actions Taken by Equifax and Federal Agencies in Response to the 2017 Breach.
- PMI (2021), Pulse of the Profession — Beyond Agility.
- RAMAČ R. et al. (2022), « Prevalence, common causes and effects of technical debt », Journal of Systems and Software, 184.
- Guide de l'achat public – Le sourcing opérationnel, Direction des achats de l'État, édition 2019.

WEBOGRAPHIE

- VERIZON (2026), Data Breach Investigations Report 2026.
- MICROSOFT, « Lifecycle Workflows », Microsoft Learn.
- MICROSOFT, « Retrieval-Augmented Generation (RAG) », Microsoft Learn.
- MCKINSEY (2012), The Social Economy: Unlocking Value and Productivity through Social Technologies.
- PANOPTO / YOUGOV (2018), « Inefficient Knowledge Sharing Costs Large Businesses $47 Million Per Year ».
- ATLASSIAN (2025), « State of Developer Experience Report 2025 ».
- ANSSI (2026), Panorama de la cybermenace 2025.
- ENISA (2025), ENISA Threat Landscape 2025.
- CIS, « CIS Control 2 — Inventory and Control of Software Assets ».
- IBM (2026), Cost of a Data Breach Report 2026.
- MICROSOFT (2023), « Will AI Fix Work? — Work Trend Index

TABLE DES FIGURES

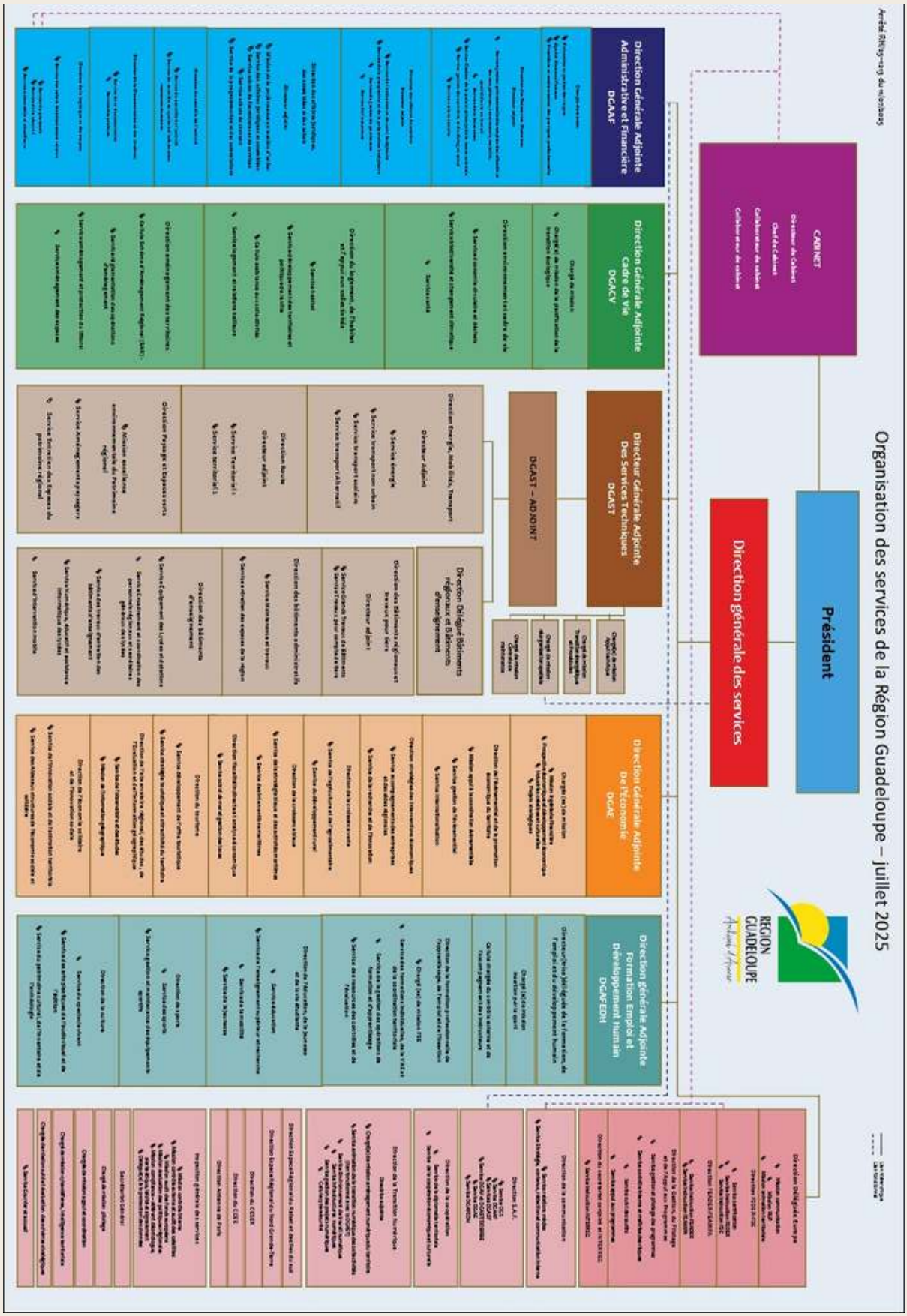
TABLE DES TABLEAUX

ANNEXES

19 PAGES

ANNEXE A — ORGANISATION DE LA COLLECTIVITÉ

FIGURE A.1 — ORGANIGRAMME DE LA RÉGION GUADELOUPE



ANNEXE B — TRAÇABILITÉ ET PLANIFICATION DES TRAVAUX

TABLEAU B.1 — MATRICE DE TRAÇABILITÉ DES TRAVAUX RÉALISÉS : BESOINS, CHOIX, LIVRABLES ET ÉTAT D'AVANCEMENT

Mission / travaux réalisés	Besoin ou situation initiale	Choix / démarche mise en œuvre	Livrables / résultats	État d'avancement
Internalisation de la plateforme Open Data CKAN	Réduire la dépendance à une solution externalisée, accroître la maîtrise de l'hébergement et étudier une alternative économiquement pertinente	Reprise de l'existant ; montée en compétence Docker/CKAN ; diagnostic par isolation progressive ; analyse TCO ; identification des conditions de reprise	Environnement repris et compris ; dysfonctionnement lié au port identifié ; arbre de diagnostic ; analyse TCO ; prérequis de reprise et recommandations de pérennisation	Annulé — blocage d'accès au serveur et dépendance à un interlocuteur unique
Cartographie initiale du patrimoine applicatif	Informations sur les applications, serveurs, ports, dépendances et criticités dispersées entre outils, documents et personnes	Consolidation rapide des données disponibles ; collecte auprès des agents ; rapprochement entre vision métier et données techniques ; représentation exploratoire en ArchiMate	Fichier maître de cartographie ; vue synthétique du patrimoine ; première représentation des dépendances applicatives	Réalisé, avec limites liées à l'exhaustivité des sources
Référentiel applicatif et suivi des versions	Répondre initialement à une demande sur la disparité des versions puis éviter de reproduire manuellement les mêmes consolidations	Exploitation des données SentinelOne ; modèle multi-niveaux ; normalisation progressive ; Human-in-the-loop ; API ; persistance PostgreSQL ; interface web ; tests et benchmarks	Référentiel applicatif ; tableau de bord ; vues familles/versions/ équipements ; file de qualification ; mécanisme de capitalisation des règles	Prototype fonctionnel et évalué — industrialisation et enrichissements à poursuivre
Normalisation et consolidation des données SentinelOne	Plus de 4 000 entrées logicielles techniquement précises mais difficilement exploitables à l'échelle du SI	Pipeline : prétraitement → règles connues → rapprochement → validation → mémorisation ; conservation de la donnée brute	Règles de normalisation ; classification famille/ édition/variante/version ; décisions réutilisables	Fonctionnel, amélioration progressive prévue
Modélisation BPMN de processus du SI	Plusieurs processus reposaient sur des pratiques implicites ou insuffisamment formalisées	Identification des acteurs, événements, validations, responsabilités et enchaînements ; modélisation BPMN	Diagrammes de processus concernant notamment les comptes/habilitations, validations applicatives et traitements d'alertes	Réalisé — certains processus constituent des bases à faire évoluer
Étude et cadrage d'un dispositif ITSM	Demandes et incidents traités notamment par courriel, avec besoin de centralisation, suivi, priorisation et traçabilité	Analyse du besoin ; définition des parcours et fonctions attendues ; maquettage ; étude/comparaison de propositions prestataires ; réflexion sur les KPI	Besoins fonctionnels ; maquettes de parcours ; critères de comparaison ; propositions d'indicateurs de suivi, priorité, délais et criticité	Étude / cadrage réalisés — déploiement ultérieur non documenté dans le mémoire

Traçabilité et planification des travaux – Tableau B.1 partie 2

Mission / travaux réalisés	Besoin ou situation initiale	Choix / démarche mise en œuvre	Livrables / résultats	État d'avancement
Création d'un point d'entrée pour la cellule cybersécurité	Cellule peu identifiable par les agents et absence de canal suffisamment explicite pour certains signalements	Création d'une adresse dédiée ; clarification du périmètre et des moyens de contact ; formalisation de la communication	Canal de signalement cybersécurité ; note interne de présentation de la cellule	Mis en place
Sensibilisation des agents au phishing et aux signalements cyber	Réduire les comportements aggravant un incident et améliorer la qualité des remontées vers la cellule cybersécurité	Formalisation de consignes simples ; version synthétique et visuelle ; articulation sensibilisation → signalement → réponse	Procédure « Que faire face à un mail suspect ? » ; supports de sensibilisation ; procédure visuelle de déclaration	Réalisé, réutilisable pour de futures campagnes
Formalisation du traitement des incidents de cybersécurité	Traitement reposant fortement sur l'expérience de l'opérateur et sur des opérations répétitives	Formalisation par gravité et scénario ; définition des vérifications, responsabilités, points de contrôle et actions correctives	Procédures côté utilisateurs et côté défenseurs ; chaîne de traitement formalisée	Réalisé, base de l'automatisation CyberAlerte
Développement de CyberAlerte — automatisation PowerShell	Réduire le temps consacré aux recherches, communications, contrôles et opérations répétitives lors d'un incident	Architecture modulaire PowerShell ; mode test isolé ; scénarios reproductibles ; gestion d'état ; journalisation ; tests Pester ; injection de défaillances	Script CyberAlerte ; fichiers de suivi maitre.csv, suivi.csv, actions.csv ; scénarios et tests ; matrice d'évaluation	Prototype fonctionnel partiellement validé en réel — certaines fonctions bloquées par les habilitations
Gouvernance et suivi des droits/habilitations	Manque de vue consolidée sur certains comptes, droits, applications, responsables et validations	Constitution d'un référentiel transitoire centralisé avant mise en œuvre d'une solution IAM plus structurée	Liste SharePoint de suivi des utilisateurs, comptes, applications, droits, responsables, validations et statuts	Mis en service comme réponse transitoire
Formalisation du cycle de vie des identités	Sécuriser arrivées, mobilités, cumuls de fonctions et départs ; limiter les droits obsolètes	Modèle Joiner–Mover–Leaver ; BPMN ; clarification demande/validation/exécution ; historisation non destructive	Processus de cycle de vie ; exigences de gouvernance ; responsabilités formalisées	Réalisé au niveau processus et exigences
Étude / sourcing d'une solution IAM / IGA	Dépasser le suivi manuel et disposer d'une gouvernance plus centralisée, traçable et automatisable	Définition de critères ; pondération ; questionnaires ; analyse des fonctionnalités et coûts ; comparaison de solutions, dont les pistes Netwrix/midPoint étudiées	Matrice d'évaluation ; comparaison des solutions ; SWOT ; analyse économique ; recommandations	Étude réalisée — démarche IAM restant à poursuivre

Traçabilité et planification des travaux – Tableau B.1 partie 3

Mission / travaux réalisés	Besoin ou situation initiale	Choix / démarche mise en œuvre	Livrables / résultats	État d'avancement
Audit Windows Hello for Business	Évaluer une évolution des mécanismes d'authentification et identifier les conditions techniques et de sécurité associées	Analyse du dispositif et de son intégration dans l'environnement Microsoft de la collectivité	Analyse / recommandations issues de l'audit	Audit réalisé — à documenter en annexe uniquement si les éléments peuvent être divulgués
Étude d'un assistant documentaire fondé sur l'IA	Difficultés d'accès, de recherche et de transmission de connaissances dispersées dans les documents	Cadrage des cas d'usage ; approche RAG ; prise en compte des droits documentaires ; architecture fonctionnelle ; analyse des risques ; présentation à la direction	Proposition structurée ; architecture ; support de présentation ; rapport de cadrage / cahier des charges initial	Arrêté au stade de l'étude — conditions de reprise identifiées
Expertise d'une proposition d'intranet WordPress	Vérifier si un POC WordPress répondait réellement au périmètre attendu d'un intranet complet	Relecture experte du périmètre et du POC ; distinction entre migration éditoriale démontrée et fonctions non validées ; analyse architecture, sécurité, exploitation et maintien en conditions opérationnelles	Fiche / avis d'expertise ; identification des écarts et risques ; points à éprouver avant généralisation	Expertise réalisée — POC jugé insuffisant à lui seul pour démontrer l'aptitude au périmètre complet
Production de procédures et documentation technique transverse	Réduire la dépendance aux connaissances individuelles et rendre les travaux transmissibles	Documentation au fil des missions ; formalisation de procédures, diagnostics, décisions, conditions de reprise et limites	Procédures d'exploitation et de sécurité ; documents de cadrage ; guides de reprise ; supports de référence	Activité continue au cours de l'apprentissage
Analyses, études comparatives et aide à la décision	Plusieurs besoins nécessitaient une analyse préalable avant choix technique ou organisationnel	Comparaisons multicritères ; TCO ; SWOT ; matrices de décision ; analyse de risques ; preuves plutôt que sélection intuitive	Études CKAN, IAM, ITSM, intranet et autres notes d'aide à la décision	Réalisé sur plusieurs missions
Appui transverse à la cellule cybersécurité et aux sujets infrastructure/réseaux	Effectif permanent réduit et besoins ponctuels variés nécessitant un soutien technique et analytique	Intervention autonome selon les priorités ; échanges avec équipes techniques, responsables de projet et prestataires ; analyse et résolution de problèmes	Contributions techniques, analyses, documentation et appui aux projets selon les besoins	Activité transverse continue

ANNEXE B — TRAÇABILITÉ ET PLANIFICATION DES TRAVAUX

TABLEAU B.2 —PLANNING PRÉVISIONNEL ET EFFECTIF DES PRINCIPALES MISSIONS ET MONTÉES EN COMPÉTENCE

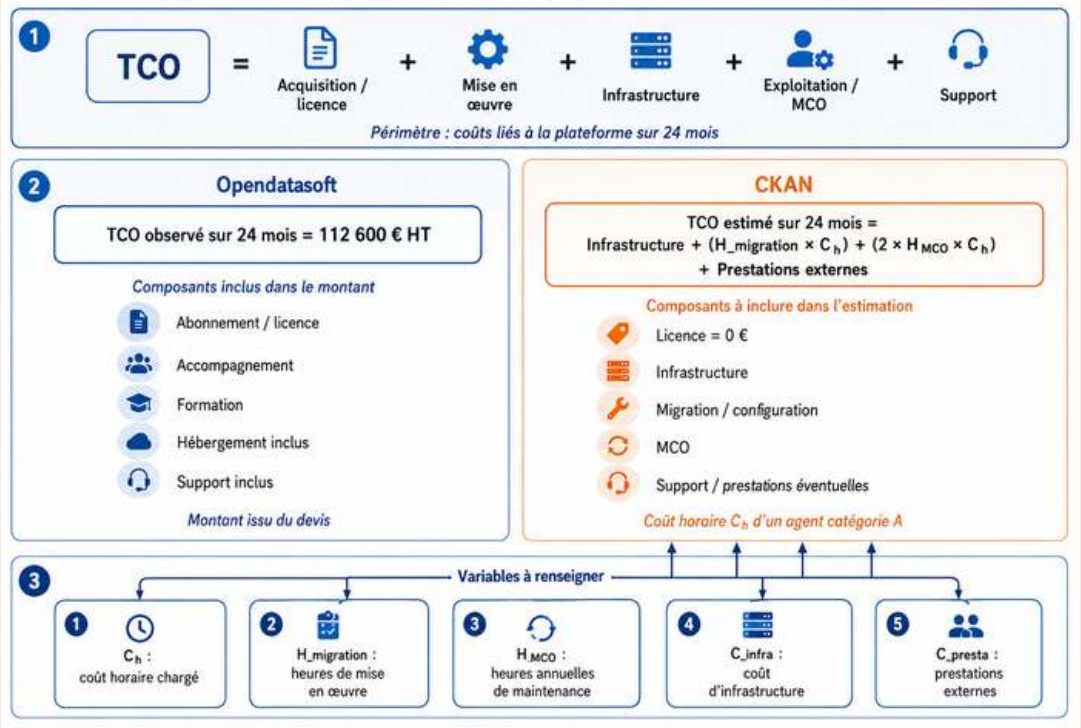
Mission / activité	Période prévisionnelle	Durée prévue	Réalisation effective	Écart / évolution
Internalisation de la plateforme CKAN	Début oct. 2024 → fin année de M1	≈ 1 an	Début oct. 2024 → déc. 2024 / janv. 2025	Projet interrompu dès le 3e mois, malgré la résolution du dysfonctionnement initial, en raison de la perte d'accès au serveur et de la dépendance à un interlocuteur unique
↳ Montée en compétence Docker	2025	≈ 35 h	Formation puis mise en pratique directement sur l'environnement CKAN	Montée en compétence réalisée en parallèle du diagnostic et de la reprise de l'existant
Première cartographie du patrimoine applicatif	13 oct. 2025 → 17 oct. 2025	5 jours ouvrés	13 oct. 2025 → 17 oct. 2025	Aucun écart significatif ; consolidation et première cartographie livrées dans le délai imposé
Formalisation et modélisation BPMN de processus	Activité transverse sur les deux années de Master	≈ 1 journée pour une première version d'un processus	2024 → août 2026, selon les besoins des différents projets	Les modèles ont été produits puis ajustés progressivement en fonction des échanges, validations et évolutions des processus
↳ Montée en compétence Python	2025	≈ 35 h	Formation suivie de mises en pratique sur les travaux d'automatisation et de traitement de données	Compétence ensuite réutilisée pour le référentiel applicatif et différents traitements
↳ Formation Power Automate	2025	7 h	Formation réalisée en 2025	Formation courte destinée à élargir les possibilités d'automatisation de processus
Étude et conception de CyberAlerte — automatisation PowerShell	À partir du 12 mars 2026	2 semaines de montée en compétence puis réalisation des premiers attendus	12 mars 2026 → août 2026	Après la montée en compétence, le premier périmètre fonctionnel a été réalisé en ≈ 8 jours. De nouvelles fonctions ont ensuite été demandées progressivement, avec en moyenne ≈ 3 jours d'implémentation par évolution

Traçabilité et planning prévisionnel des travaux – Tableau B.2 partie 2

Mission / activité	Période prévisionnelle	Durée prévue	Réalisation effective	Écart / évolution
↳ Montée en compétence PowerShell	12 mars 2026 → fin mars 2026	≈ 2 semaines / ≈ 35 h	Réalisée au démarrage de CyberAlerte, parallèlement aux premières expérimentations	Acquisition directement orientée vers les commandes Microsoft et les besoins du script
Étude d'un assistant documentaire fondé sur l'IA	10 mars 2026 → 18 mai 2026	≈ 2 mois	10 mars → 18 mai 2026	Les travaux techniques ont été interrompus environ deux semaines après le lancement ; le cadrage a néanmoins été poursuivi suffisamment pour finaliser le cahier des charges et les artefacts de conception
↳ Montée en compétence RAG / IA générative	2026	≈ 35 h	Formation et recherches réalisées en parallèle du cadrage de l'assistant documentaire	Les connaissances acquises ont été mobilisées pour définir l'architecture RAG, les contraintes documentaires et les mécanismes de contrôle d'accès
Gouvernance des identités et des accès / étude IAM	≈ printemps–été 2026*	≈ 1 à 2 mois*	≈ mai → juil. 2026*	Le besoin initial de formalisation des habilitations s'est élargi vers le cycle de vie Joiner–Mover–Leaver, la traçabilité, la comparaison de solutions et l'étude d'une réponse IAM structurée
Second livrable de cartographie applicative	15 juin 2026 → 26 juin 2026	2 semaines	15 juin → 26 juin 2026	Livrable demandé réalisé dans le délai initial
Extension vers un référentiel applicatif automatisé	À partir du livrable de juin 2026	Extension initialement non prévue ; ≈ 2 mois finalement consacrés	Fin juin → août 2026	Le besoin ponctuel de cartographie a été volontairement étendu vers une application pérenne : collecte, normalisation, référentiel, interface et évaluations
↳ Approfondissement Python / traitement de données	2026, au fil du développement	Selon besoin	Mobilisation continue pendant le développement du référentiel	Apprentissage directement guidé par les problématiques de normalisation, API, persistance et tests

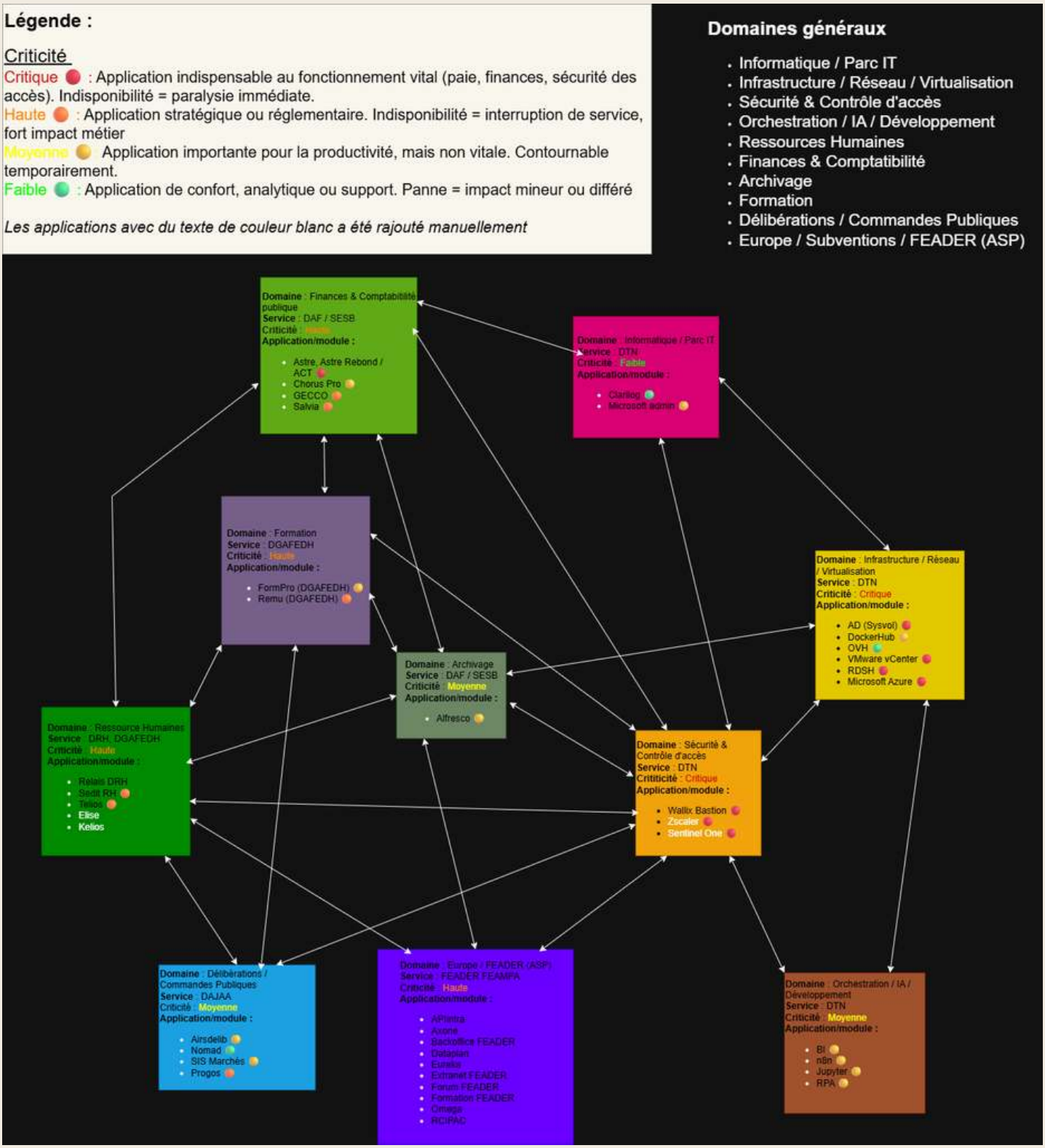
ANNEXE C — INTERNALISATION DE CKAN : ANALYSE ÉCONOMIQUE

FIGURE C.1 — STRUCTURE SIMPLIFIÉE DU TCO COMPARATIF OPENDATASOFT/CKAN



ANNEXE D — CARTOGRAPHIE ET RÉFÉRENTIEL APPLICATIF

FIGURE D.1 — VUE ARCHIMATE DE LA CARTOGRAPHIE APPLICATIVE



ANNEXE D — CARTOGRAPHIE ET RÉFÉRENTIEL APPLICATIF

TABLEAU D.1 — FONCTIONS PRINCIPALES DE LA V0 DU RÉFÉRENCIEL APPLICATIF

Fonction	Modules principaux	Logique appliquée
Import CSV	ImportService, csv_validation, parsers inventaire/risques	Vérifie l'extension, l'encodage UTF-8 et le contrat de colonnes, puis parse chaque ligne et conserve les valeurs brutes valides.
Historisation des imports	ImportBatch, ImportBatchRepository	Crée un lot par fichier, avec la source, les compteurs, les avertissements et les erreurs.
Création d'analyse	AnalysisRunService	Relie explicitement un lot d'inventaire et/ou de risques dans un périmètre fonctionnel.
Normalisation	NormalizationService, NormalizationRule	Transforme une paire brute nom/vendor en résultat tracé, en appliquant en priorité les règles valides.
Consolidation	ApplicationConsolidationService	Crée ou retrouve l'application consolidée, puis y rattache les lignes brutes.
Rapprochement risques/inventaire	RiskInventoryMatchingService	Rattache un risque à une application consolidée lorsque le résultat déterministe le permet.
Qualification	QualificationService, StatusCalculationService	Distingue le statut de contrôle des statuts techniques de normalisation et de matching.
Priorisation	PriorityService, RecommendationService	Calcule une priorité de P1 à P5 ainsi qu'une recommandation unique et explicable, à partir des données agrégées.
Agrégats	RiskAggregationService	Crée une ligne par couple AnalysisRun + Application, servant de base aux KPI, aux listes et aux exports.
Consultation	DashboardService, ListQueryService, ApplicationDetailService	Lit une analyse résolue, avec pagination, filtres et tri côté serveur.
Familles	ApplicationFamilyAggregationService	Regroupe les agrégats par code de famille, sans recalculer le scoring ni relire les CSV bruts.
Export Excel	ExcelExportService, feuilles spécialisées, ExportFileStorage	Produit un fichier synchrone, nommé et stocké côté serveur, à partir d'un analysis_run_id explicite.

ANNEXE D — CARTOGRAPHIE ET RÉFÉRENTIEL APPLICATIF

FIGURE D.2 — PRÉSERVATION DE LA DONNÉE BRUTE ET APPLICATION CONTRÔLÉE DES RÈGLES DE NORMALISATION

```
def _clean_value(value: str) -> str:
    return " ".join(value.strip().split())


def _match_key(value: str) -> str:
    return _clean_value(unescape(value)).casefold()
```

```
rules = tuple(
    session.scalars(
        select(NormalizationRule)
        .where(NormalizationRule.validated.is_(True))
        .order_by(NormalizationRule.id)
    ).all()
)
if not rules:
    return cls()
```

Les valeurs utilisées pour le rapprochement sont nettoyées et homogénéisées sans modifier les données brutes issues de SentinelOne. Les règles persistées en base ne sont par ailleurs chargées automatiquement que lorsqu'elles ont été explicitement validées (validated=True), ce qui évite qu'une proposition encore en attente de qualification influence silencieusement les traitements suivants.

FIGURE D.3 — GARDE-FOUS CONTRE LES RAPPROCHEMENTS TROP LARGES

```
exact_match = rules.exact.get(value_key)
if exact_match is not None and _vendor_guard_matches(exact_match, normalized_vendor):
    return exact_match

for prefix_match in rules.prefixes:
    if _prefix_matches(value_key, _match_key(prefix_match.raw_pattern)):
        if _vendor_guard_matches(prefix_match, normalized_vendor):
            return prefix_match
```

Le moteur privilégie une correspondance exacte avant d'envisager une règle par préfixe. Dans les deux cas, le fournisseur attendu constitue un contrôle supplémentaire. Les règles par préfixe sont également ordonnées du motif le plus spécifique au plus général et ne s'appliquent qu'à une frontière de mot reconnue. Cette stratégie vise à réduire le risque de faux regroupements, particulièrement problématiques dans la perspective d'un rapprochement ultérieur avec des vulnérabilités.

FIGURE D.4 — TEST DE NON-RÉGRESSION : PRÉSERVER LE BRUT TOUT EN NORMALISANT

```
result = service.normalize("Adobe Photoshop 2024", "Adobe Inc.")

assert result.raw_application_name == "Adobe Photoshop 2024"
assert result.normalized_application_name == "Adobe Photoshop"
assert result.raw_vendor == "Adobe Inc."
assert result.normalized_vendor == "Adobe"
assert result.application_family == "multimedia"
```

FIGURE D.5 — CONFIRMATION EXPLICITE REQUISE POUR UNE RÈGLE PREFIX

```
with pytest.raises(NormalizationRuleCorrectionError, match="confirmation explicite"):
    NormalizationRuleCorrectionService(db_session).create_correction(
        _prefix_request(confirmed=False)
    )
```

FIGURE D.6 — ARCHITECTURE FONCTIONNELLE DU RÉFÉRENCIEL APPLICATIF

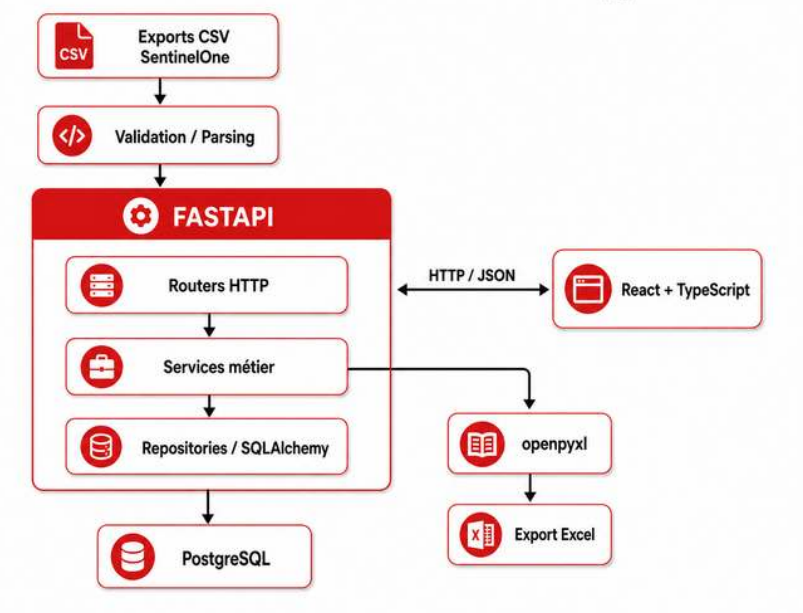
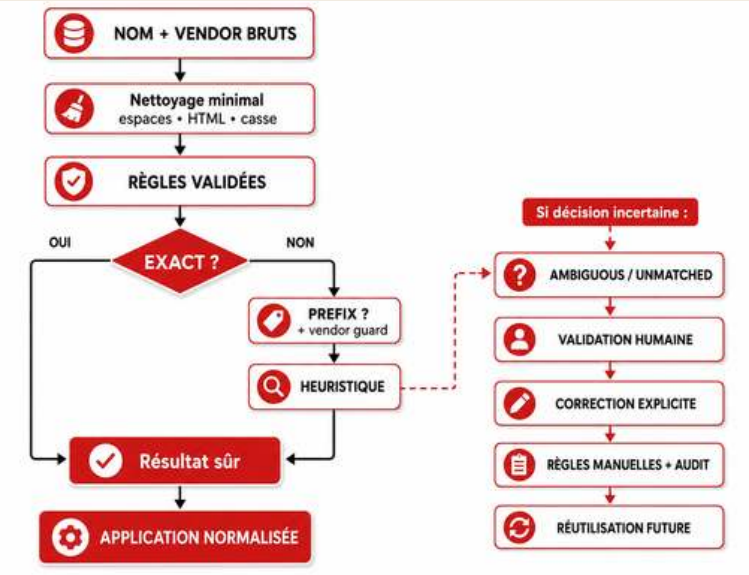


FIGURE D.7 — PIPELINE DE NORMALISATION



ANNEXE E — CYBERALERTE : PROCÉDURES, TESTS ET INDICATEURS

FIGURE E.1 — VISUEL DE COMMUNICATION POUR LA SENSIBILISATION DES AGENTS

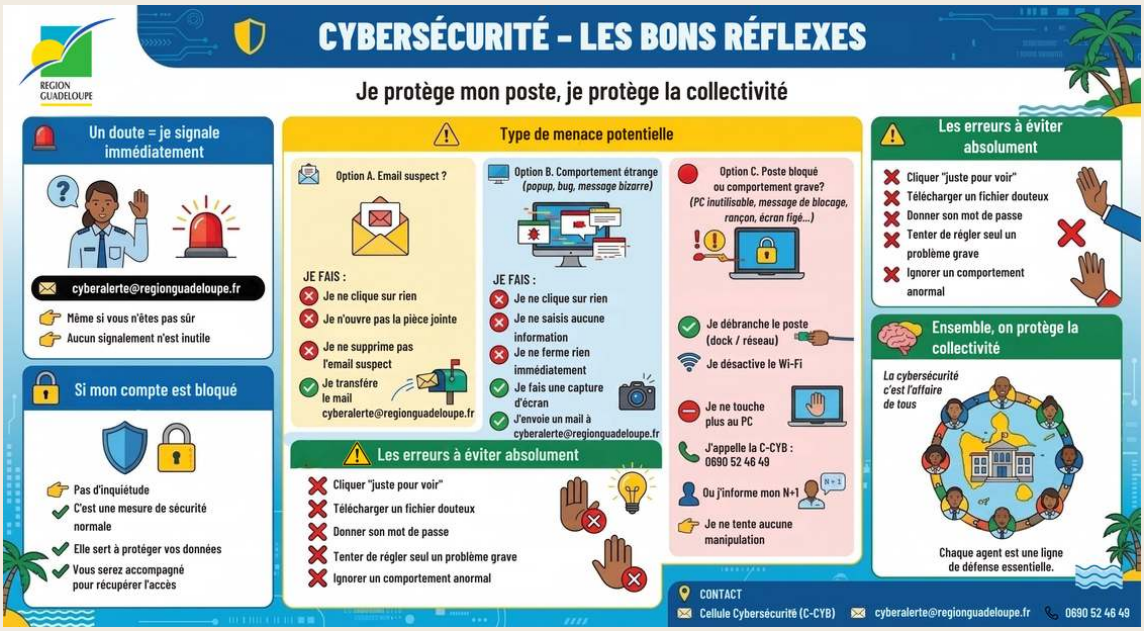


FIGURE E.2 — EXEMPLES DE KPI DE CYBERSÉCURITÉ ET DE GOUVERNANCE DU SI (VALEURS FICTIVES)

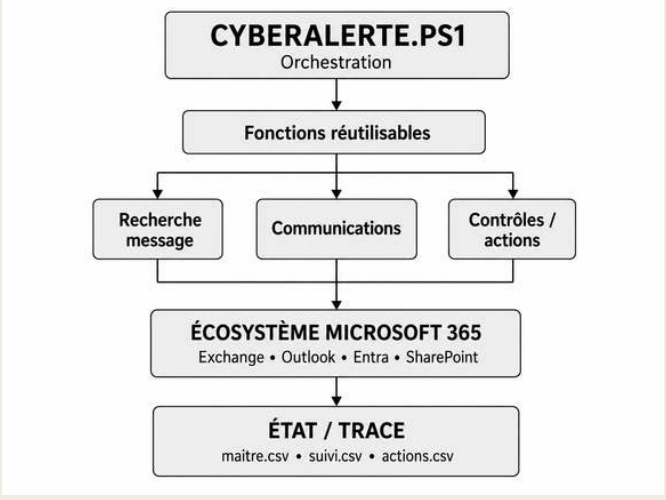
Indicateur	Semaine N	Semaine N-1	Évolution
Tentatives phishing	42	30	+40%
Mails malveillants bloqués	320	280	+14%
Phishing signalé par agents	8	5	+60%
Utilisateur à risque (GP, MQ, GF, FR, MF)	8	4	+50%

Indicateur	Semaine N	Semaine N-1	Évolution
Incidents cybersécurité	6	4	+50%
Incidents critiques	1	0	+1

Indicateur	Semaine N	Semaine N-1	Évolution
Vulnérabilités détectées	20	25	-20%
Vulnérabilités corrigées	15	10	+50%

Indicateur	Semaine N	Semaine N-1
Comptes créés	12	10
Comptes supprimés	8	5

FIGURE E.3 — ARCHITECTURE FONCTIONNELLE DE CYBERALERTE



ANNEXE F — GOUVERNANCE DES IDENTITÉS ET DES ACCÈS

FIGURE F.1 — PROCESSUS BPMN DU CYCLE DE VIE DES IDENTITÉS ET DES HABILITATIONS

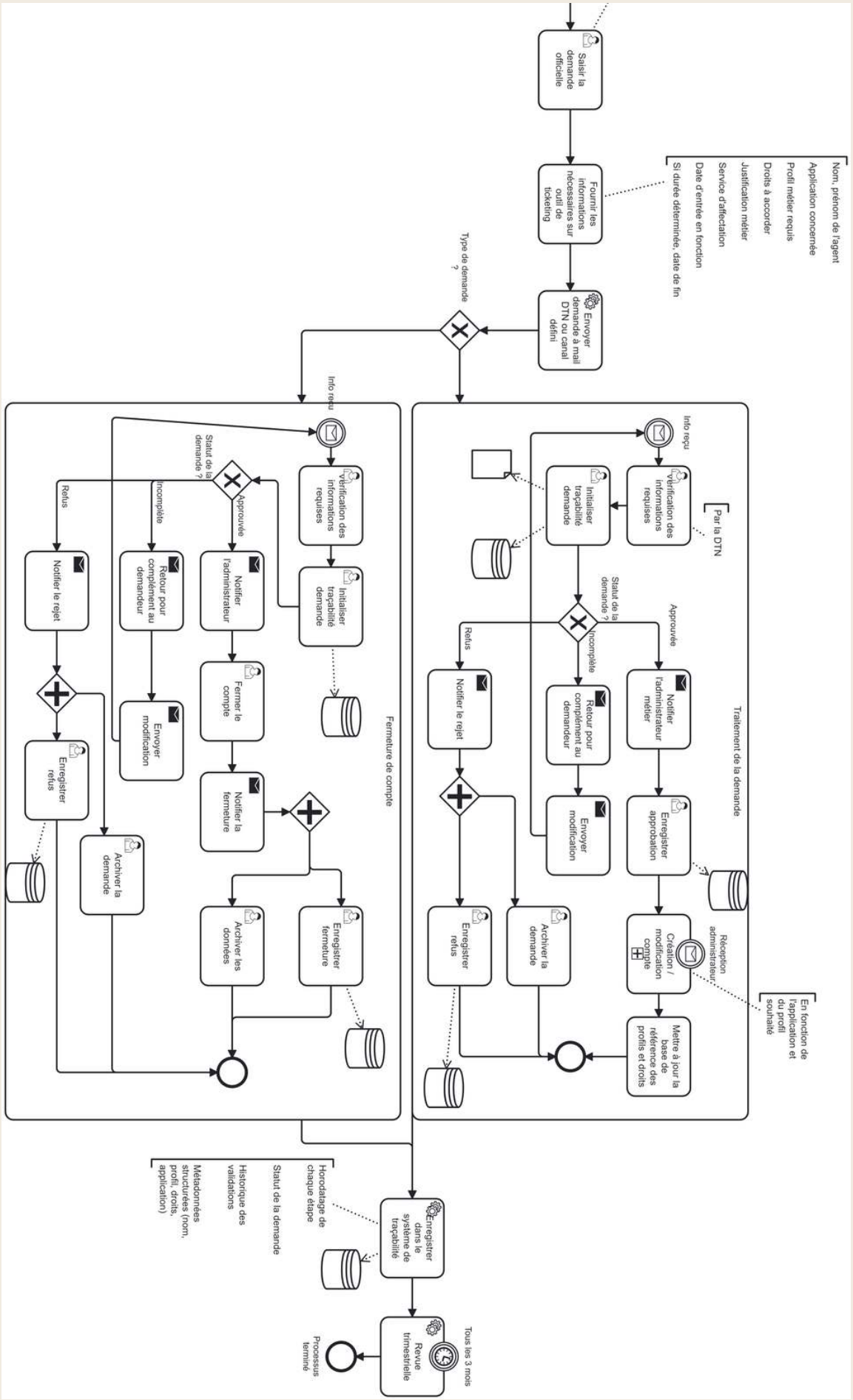
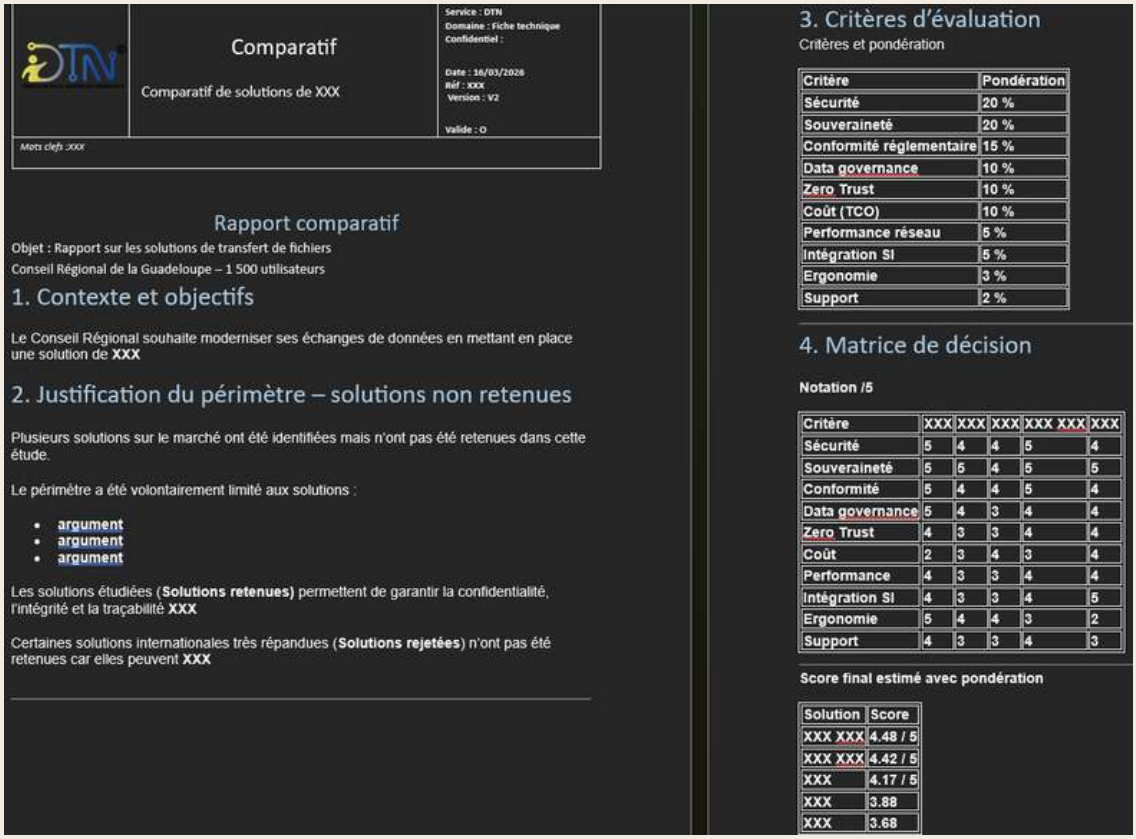


FIGURE F.2 — CRITÈRES ET PONDÉRATIONS DE LA MATRICE D'ÉVALUATION DES SOLUTIONS IAM



DTN

Comparatif

Comparatif de solutions de XXX

Service : DTN
Domaine : Fiche technique
Confidentiel :

Date : 16/03/2026
Réf : XXX
Version : V2

Valide : O

Mots clefs :XXX

Rapport comparatif

Objet : Rapport sur les solutions de transfert de fichiers
Conseil Régional de la Guadeloupe – 1 500 utilisateurs

1. Contexte et objectifs

Le Conseil Régional souhaite moderniser ses échanges de données en mettant en place une solution de **XXX**

2. Justification du périmètre – solutions non retenues

Plusieurs solutions sur le marché ont été identifiées mais n'ont pas été retenues dans cette étude.

Le périmètre a été volontairement limité aux solutions :

- **argument**
- **argument**
- **argument**

Les solutions étudiées (**Solutions retenues**) permettent de garantir la confidentialité, l'intégrité et la traçabilité **XXX**

Certaines solutions internationales très répandues (**Solutions rejetées**) n'ont pas été retenues car elles peuvent **XXX**

3. Critères d'évaluation

Critères et pondération

Critère	Pondération
Sécurité	20 %
Souveraineté	20 %
Conformité réglementaire	15 %
Data governance	10 %
Zero Trust	10 %
Coût (TCO)	10 %
Performance réseau	5 %
Intégration SI	5 %
Ergonomie	3 %
Support	2 %

4. Matrice de décision

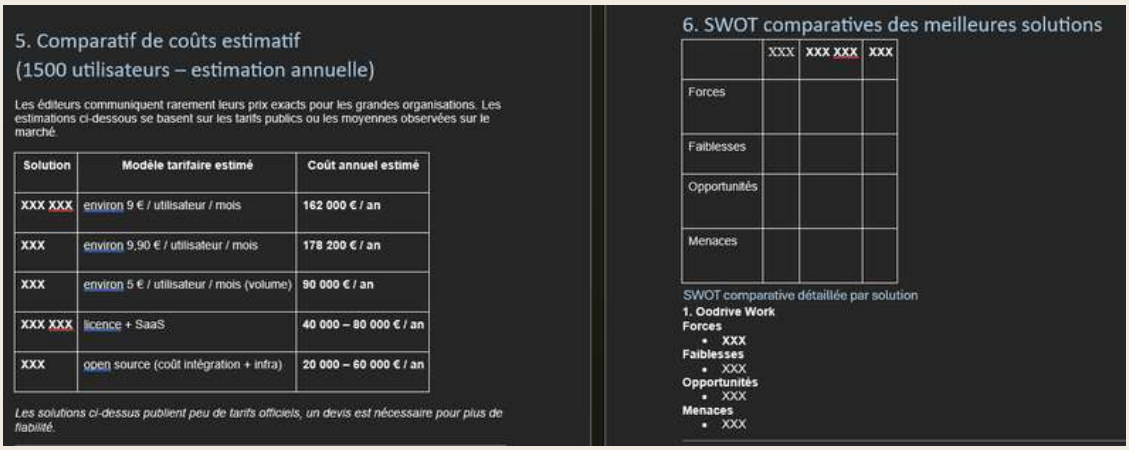
Notation /5

Critère	XXX	XXX	XXX	XXX XXX	XXX
Sécurité	5	4	4	5	4
Souveraineté	5	5	4	5	5
Conformité	5	4	4	5	4
Data governance	5	4	3	4	4
Zero Trust	4	3	3	4	4
Coût	2	3	4	3	4
Performance	4	3	3	4	4
Intégration SI	4	3	3	4	5
Ergonomie	5	4	4	3	2
Support	4	3	3	4	3

Score final estimé avec pondération

Solution	Score
XXX XXX	4.48 / 5
XXX XXX	4.42 / 5
XXX	4.17 / 5
XXX	3.88
XXX	3.68

FIGURE F.3 — COMPARATIF ESTIMATIF DES COÛTS ET ANALYSE SWOT



5. Comparatif de coûts estimatif (1500 utilisateurs – estimation annuelle)

Les éditeurs communiquent rarement leurs prix exacts pour les grandes organisations. Les estimations ci-dessous se basent sur les tarifs publics ou les moyennes observées sur le marché.

Solution	Modèle tarifaire estimé	Coût annuel estimé
XXX XXX	environ 9 € / utilisateur / mois	162 000 € / an
XXX	environ 9,90 € / utilisateur / mois	178 200 € / an
XXX	environ 5 € / utilisateur / mois (volume)	90 000 € / an
XXX XXX	licence + SaaS	40 000 – 80 000 € / an
XXX	open source (coût intégration + infra)	20 000 – 60 000 € / an

Les solutions ci-dessus publient peu de tarifs officiels, un devis est nécessaire pour plus de fiabilité.

6. SWOT comparatives des meilleures solutions

	XXX	XXX XXX	XXX
Forces			
Faiblesses			
Opportunités			
Menaces			

SWOT comparative détaillée par solution

1. Oodrive Work

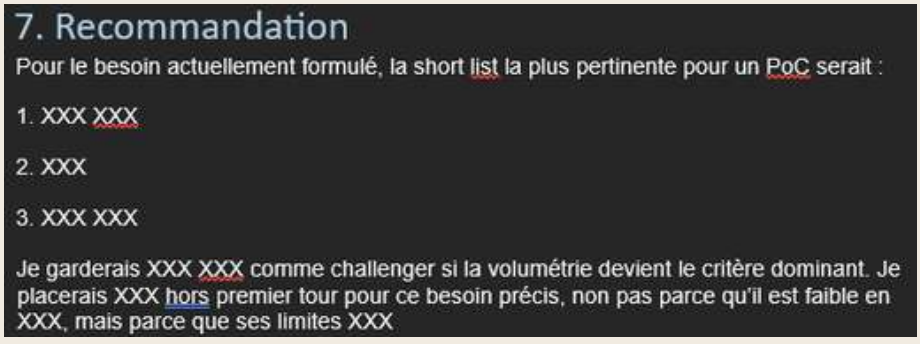
Forces
- XXX

Faiblesses
- XXX

Opportunités
- XXX

Menaces
- XXX

FIGURE F.4 — RECOMMANDATION FINALE DU COMPARATIF



7. Recommandation

Pour le besoin actuellement formulé, la short list la plus pertinente pour un PoC serait :

1. XXX XXX
2. XXX
3. XXX XXX

Je garderais XXX XXX comme challenger si la volumétrie devient le critère dominant. Je placerais XXX hors premier tour pour ce besoin précis, non pas parce qu'il est faible en XXX, mais parce que ses limites XXX

FIGURE F.5 — EXEMPLE GUIDE DE SOURCING RÉALISÉ

Documents et preuves à demander - sécurité de la messagerie

ID	Domaine	Document demandé	Niveau	Statut prestataire 1	Date de demande	Date de réception	Lien
DOC01	Fournisseur	Présentation juridique, capitalistique, organigramme et chiffres clés	Essentiel	Non demandé			
DOC02	Fournisseur	Références clients comparables, notamment secteur public / collectivités	Essentiel	Non demandé			
DOC03	Produit	Matrice fonctionnelle détaillée de l'offre et des éditions/options	Essentiel	Non demandé			
DOC04	Produit	Roadmap produit sur trois ans	Utile	Non demandé			
DOC05	Détection	Description des moteurs de détection, chaîne de filtrage et threat intelligence	Essentiel	Non demandé			
DOC06	Détection	Métriques de détection, faux positifs/négatifs et méthodologie de mesure	Essentiel	Non demandé			
DOC07	Architecture	Architecture SaaS / on-premises / hybride et haute disponibilité	Essentiel	Non demandé			
DOC08	Architecture	Matrice des flux réseau, IP, ports, certificats et prérequis DNS	Essentiel	Non demandé			
DOC09	Intégration	Guide d'intégration Microsoft 365 / Exchange	Essentiel	Non demandé			
DOC10	Intégration	Guide synchronisation Entra ID / LDAP	Essentiel	Non demandé			
DOC11	Intégration	Documentation API, webhooks, SIEM/syslog et formats d'événements	Essentiel	Non demandé			
DOC12	Sécurité	Plan d'assurance sécurité / politique de sécurité	Essentiel	Non demandé			
DOC13	Sécurité	Certifications et attestations avec périmètres de validité	Essentiel	Non demandé			
DOC14	Sécurité	Synthèse récente de test d'intrusion / audit indépendant	Essentiel	Non demandé			
DOC15	Sécurité	Politique de gestion des vulnérabilités, délais et SBOM	Essentiel	Non demandé			
DOC16	Continuité	PCA/PRA, RPO/RTO et preuve de tests	Essentiel	Non demandé			
DOC17	Données	Cartographie des données, hébergeurs, sous-traitants et localisations	Essentiel	Non demandé			
DOC18	Données	DPA / clauses RGPD et politiques de conservation / suppression	Essentiel	Non demandé			
DOC19	Projet	Méthodologie de déploiement, planning, RACI et runbook de migration	Essentiel	Non demandé			
DOC20	Projet	Protocole POC / pilote et critères de réussite	Essentiel	Non demandé			
DOC21	RUN	Catalogue de support, horaires, SLA et matrice d'escalade	Essentiel	Non demandé			
DOC22	RUN	Exemple de rapport de service / statistiques / cockpit	Utile	Non demandé			
DOC23	Économie	Grille de licences, éditions, options, services et TCO 5 ans	Essentiel	Non demandé			
DOC24	Contrat	Conditions de renouvellement, indexation et résiliation	Essentiel	Non demandé			
DOC25	Réversibilité	Plan de réversibilité et formats d'export	Essentiel	Non demandé			
DOC26	Juridique	Matrice de propriété intellectuelle des spécifiques	Utile	Non demandé			
DOC27	Assurance	Attestations RC professionnelle et cyber	Utile	Non demandé			
DOC28	Sensibilisation	Présentation du bouton phishing, simulations et traitement d'un signalement réel	Utile	Non demandé			
DOC29	RSE	Politique RSE / environnement / insertion	Utile	Non demandé			

01_Cadrage | 02_Questionnaire_Email | 03_Entretien_Maillnblack | 04_Synthese_prestataires | 05_Documents_attendus | 06_Scenarios_demo | 07_Invitation_type | 08_Sources | 99_Listes

ANNEXE G — ASSISTANT DOCUMENTAIRE FONDÉ SUR L'IA

FIGURE G.1 — SUPPORT SYNTHÉTIQUE DE PRÉSENTATION DE LA PROPOSITION

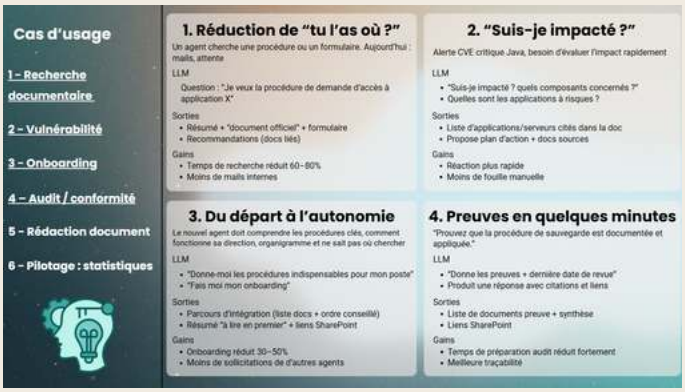


FIGURE G.2 — PRÉREQUIS ISSUS DU CAHIER DES CHARGES EN MATIÈRE D'ARCHITECTURE TECHNIQUE ET DE RESSOURCES

ANNEXE H — CADRE D'ANALYSE DE LA DETTE ORGANISATIONNELLE

FIGURE H.1 — BANQUE DE QUESTIONS DE DÉTECTION DES FRICTIONS ORGANISATIONNELLES

Questions transversales — Détection initiale
Q01
Est-ce que je perds régulièrement du temps à retrouver une information ?
Q02
Une activité importante dépend-elle d'une seule personne ?
Q03
Une solution initialement temporaire est-elle devenue permanente ?
Q04
Quelles tâches dois-je régulièrement refaire ou contourner ?
Q05
Qu'est-ce qui me fait perdre du temps de manière répétée ?
Q06
Une difficulté considérée aujourd'hui comme « normale » pourrait-elle résulter d'un ancien compromis ?
Q07
Existe-t-il une opération simple qui demande aujourd'hui beaucoup plus d'étapes qu'elle ne devrait ?
Q08
Un problème déjà rencontré réapparaît-il régulièrement sans être réellement résolu ?
Q09
Certaines activités fonctionnent-elles uniquement grâce à des habitudes ou connaissances informelles ?
Q10
Une décision prise pour gagner du temps continue-t-elle à produire des contraintes alors que l'urgence initiale a disparu ?

Information et documentation
Q11
Les informations nécessaires à une activité sont-elles faciles à retrouver ?
Q12
Existe-t-il plusieurs documents ou fichiers contenant des versions différentes de la même information ?
Q13
Est-il difficile de déterminer quelle documentation constitue la référence actuelle ?
Q14
Certaines configurations ou décisions importantes ne sont-elles connues que par les personnes qui les ont réalisées ?
Q15
La documentation décrit-elle fidèlement l'état réel des systèmes, processus ou configurations ?
Q16
Une modification peut-elle être réalisée sans que sa documentation soit ensuite mise à jour ?
Q17
Faut-il régulièrement solliciter plusieurs personnes pour reconstruire une information qui devrait être disponible directement ?
Q18
Une information critique risque-t-elle d'être perdue lors du départ ou de l'indisponibilité d'un collaborateur ?

Processus
Q19
Les étapes nécessaires à la réalisation de l'activité sont-elles clairement définies ?
Q20
Une même tâche est-elle exécutée différemment selon la personne qui la réalise ?
Q21
Certaines étapes manuelles sont-elles répétées alors qu'elles pourraient être standardisées ou automatisées ?
Q22
Des validations sont-elles régulièrement retardées parce que le responsable ou le circuit de décision n'est pas clairement identifié ?
Q23
Des contournements sont-ils devenus nécessaires pour terminer normalement une activité ?
Q24
Les mêmes données doivent-elles être ressaisies ou retraitées dans plusieurs outils ?
Q25
Un incident ou une demande nécessite-t-il régulièrement de reconstruire la procédure à suivre ?
Q26
Certaines exceptions se sont-elles accumulées au point de devenir une partie normale du processus ?

Humain et compétences
Q27
Une activité critique ne peut-elle être réalisée que par une seule personne ou un nombre très limité de personnes ?
Q28
L'absence d'un collaborateur suffit-elle à retarder ou bloquer un processus ?
Q29
Certaines compétences indispensables sont-elles insuffisamment partagées dans l'équipe ?
Q30
La transmission des connaissances dépend-elle principalement d'échanges oraux ou informels ?
Q31
Un nouvel intervenant aurait-il des difficultés importantes à reprendre l'activité avec les seuls éléments disponibles ?
Q32
Certaines opérations nécessitent-elles systématiquement l'intervention de la même personne parce qu'elle seule connaît l'historique du sujet ?
Q33
L'organisation dispose-t-elle d'une solution de continuité lorsqu'une compétence clé devient indisponible ?
Q34
Des tâches à faible valeur ajoutée mobilisent-elles régulièrement des compétences rares ou spécialisées ?

Outils et architecture
Q35
Une solution est-elle encore utilisée principalement parce qu'elle est historiquement en place ?
Q36
Une modification mineure nécessite-t-elle aujourd'hui des adaptations importantes sur plusieurs composants ?
Q37
Des configurations spécifiques ou exceptions techniques se sont-elles accumulées au fil du temps ?
Q38
Existe-t-il plusieurs versions, variantes ou outils remplissant des fonctions proches sans justification clairement établie ?
Q39
Certaines opérations nécessitent-elles des manipulations manuelles entre plusieurs outils qui communiquent mal entre eux ?
Q40
L'organisation connaît-elle précisément les dépendances nécessaires au fonctionnement de ses applications critiques ?
Q41
La maintenance d'un système est-elle rendue difficile par des choix techniques anciens ou insuffisamment documentés ?
Q42
Une solution temporaire, un prototype, un fichier Excel ou un script est-il devenu un composant indispensable du fonctionnement courant ?

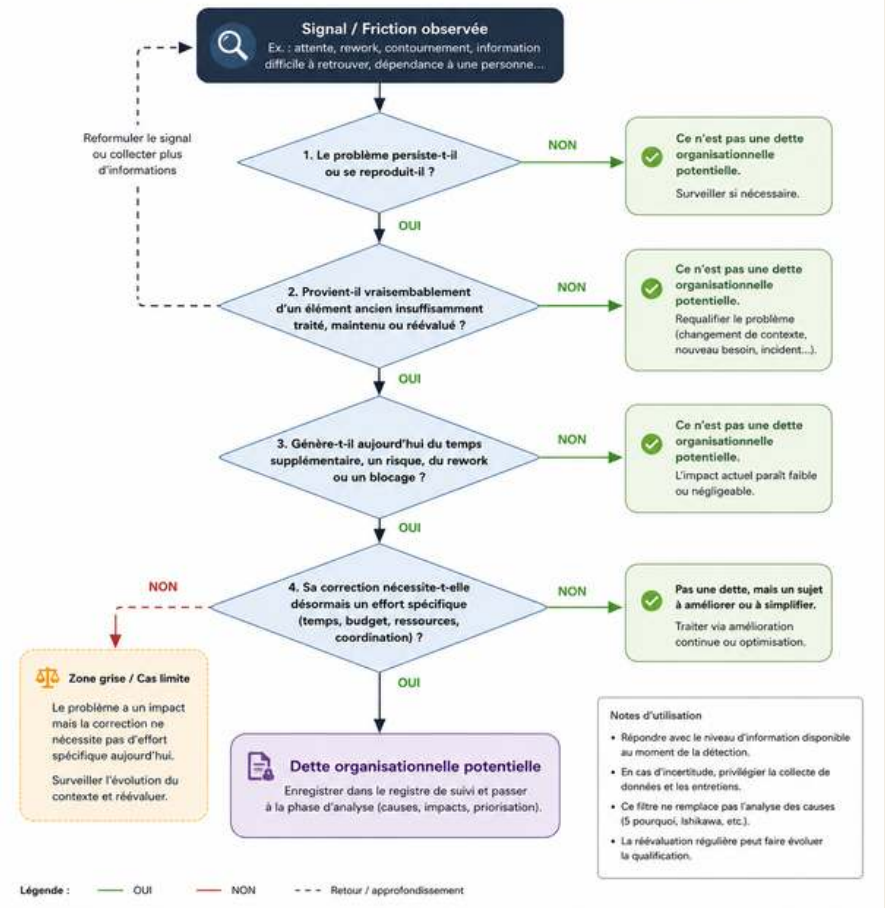
Gouvernance et responsabilités

Q43
Est-il toujours possible d'identifier qui est responsable d'une application, d'une donnée, d'un droit ou d'un processus ?
Q44
Est-il possible de savoir qui a pris ou validé une décision importante et pour quelle raison ?
Q45
Les droits et responsabilités sont-ils régulièrement réévalués lorsque les personnes ou les besoins évoluent ?
Q46
Certaines décisions restent-elles en attente faute d'un acteur clairement responsable de leur arbitrage ?
Q47
Existe-t-il des actifs ou processus pour lesquels personne ne se considère réellement responsable ?
Q48
Les décisions temporaires disposent-elles d'une échéance, d'un responsable ou d'une condition explicite de réévaluation ?
Q49
Une exception accordée ponctuellement peut-elle rester en place sans mécanisme permettant de la détecter ultérieurement ?
Q50
L'organisation peut-elle expliquer a posteriori pourquoi un choix, une habilitation ou une configuration existe encore ?

Historique et contraintes

Q51
La situation actuelle résulte-t-elle d'une urgence, d'un manque de temps, de budget ou de ressources passé ?
Q52
La contrainte qui avait justifié le compromis existe-t-elle toujours aujourd'hui ?
Q53
Une décision ancienne continue-t-elle à orienter les choix actuels alors que le contexte a changé ?
Q54
Un projet ultérieur a-t-il été construit sur une solution qui devait initialement être provisoire ?
Q55
Corriger aujourd'hui la situation demanderait-il beaucoup plus d'efforts que lorsqu'elle est apparue ?
Q56
Des systèmes, processus ou habitudes ont-ils progressivement été construits autour de cette situation ?
Q57
Des tentatives précédentes de correction ont-elles été abandonnées, reportées ou contournées ?
Q58
Le maintien de l'existant est-il devenu l'argument principal pour continuer à le conserver ?

FIGURE H.2 — ARBRE DE PRÉQUALIFICATION D'UNE DETTE ORGANISATIONNELLE POTENTIELLE



CLAUSE DE NON-CONFIDENTIALITÉ

Je soussigné, CELESTE Thierry, agissant en qualité de chef de la cellule Cybersécurité au sein de la Région Guadeloupe, certifie avoir pris connaissance du présent mémoire de Master 2 MIAGE intitulé :

« La dette organisationnelle : quand les raccourcis d'aujourd'hui deviennent les freins de demain »

rédigé par Milan LADAMUS, étudiant en Master 2 MIAGE à l'Université des Antilles.

Après vérification de son contenu, j'autorise la consultation, la présentation et la diffusion de ce mémoire dans le cadre universitaire, celui-ci ne contenant aucune information considérée comme confidentielle par la Région Guadeloupe.

Fait au Raizet, le 27 / 08 / 2026.

Le maître d'apprentissage	**L'étudiant-apprenti**
Thierry CELESTE	Milan LADAMUS
Signature : 	Signature : 

Cachet de la Collectivité Régionale de Guadeloupe :

2024 - 2026

Throughout my MIAGE course at the University of the Antilles, my approach to information systems has gradually evolved. Whilst IT may initially be viewed through the lens of development, infrastructure or tools, the MIAGE programme encourages students to consider an information system as a whole: its technical components, but also the processes, data, stakeholders, decisions and the relationships that link them.

This approach was put into practice throughout my three-year apprenticeship at the Regional Council of Guadeloupe, and more particularly during my two years of the MIAGE Master's programme, when I worked as an Infrastructure and Network Supervisor. The cross-functional nature of this role led me to tackle a wide range of issues at the intersection of development, infrastructure, cybersecurity and information system governance. Above all, it taught me that a technical solution cannot be fully understood in isolation from the organisational environment in which it is to be implemented, used, maintained and passed on.

This dissertation represents the culmination of this journey. It draws primarily on the work carried out within the local authority and seeks to document not only the outcomes, but also the process itself: the needs identified, the constraints encountered, the trade-offs made, the solutions devised, their results and their limitations. Particular attention is paid to the continuity of the work undertaken and to the conditions that enable the solutions implemented to remain understandable, maintainable and reusable beyond their immediate context.

Finally, stepping back from these experiences feeds into the topic explored in this dissertation: organizational debt. More broadly, I examine how day-to-day constraints shape decisions around information systems and how these decisions can continue to have consequences long after their implementation.

REGION GUADELOUPE

ABSTRACT

Milan LADAMUS
Superviseur Infrastructure et Réseaux